PICTURE SYSTEM 2
Maintenance Manual
Volume 2
Line Generator and Character Generator

1st Edition
2nd Edition

COPYRIGHT EVANS & SUTHERLAND COMPUTER CORPORATION
1977, 1978

Customer Engineering Dept.
Evans & Sutherland Computer Corporation
580 Arapeen Drive
Salt Lake City, Utah 84108

First Edition July 1977
Second Edition July 1978

"All information contained herein together with all drawings, diagrams and specifications herein or attendant hereto are, and remain, the property of Evans & Sutherland Computer Corporation. Many of the intellectual and technical concepts described herein are proprietary to Evans & Sutherland and may be covered by U.S. and Foreign Patents or Patents Pending or are protected as trade secrets. Any dissemination of this information or reproduction of this material for commercial or other purposes other than the express purpose for which it has been made available are strictly forbidden unless prior written permission is obtained from Evans & Sutherland Computer Corporation".

Evans & Sutherland Computer Corporation assumes no responsibility for any errors that may appear in this manual. The information in this document is subject to change without notice.

TABLE OF CONTENTS

1.0 Introduction.....1-1

2.0 The Line Generator Computation.....2-1

2.1 The Mathematics of Line Generation.....2-1
2.2 Converting to the Time Domain.....2-4
2.3 Computational Tasks.....2-6
2.4 The Line Generator Data Flow.....2-8

2.4.1 The Input FIFO's.....2-8
2.4.2 X and Y Shifter Adders.....2-13
2.4.3 XY Multiplier.....2-15
2.4.4 The Square Root Calculations.....2-16
2.4.5 The X-Y Analog Function.....2-17
2.4.6 Scope Drive, Contrast, Color Drive and Unblank..2-18

3.0 The Line Generator Control State Machine.....3-1

3.1 The Four State Machine Controllers.....3-1
3.2 The Input Control State Machine (195218).....3-5
3.3 The Arithmetic Control State Machine (195216).....3-18
3.4 The Multiplier Control State Machine (195216).....3-35
3.5 The Output Control State Machine (195213).....3-50
3.6 State Machine Controller Intercommunication (195216)...3-63

4.0 Detailed Descriptions of the Line Generator Cards.....4-1

4.1 211 Scope Driver.....4-1
4.2 214 X-Y Multiplier Card.....4-16
4.3 215 Line Generator Root Calculator.....4-20
4.4 217 X-Y Adder Shifter Card.....4-28
4.5 212 X-Y Analog.....4-37
4.6 219 Line Generator FIFO Port.....4-51
4.7 216 Arithmetic Control.....
4.8 218 Line Generator Input Control.....
4.9 213 Output Control.....
4.10 231 Z Depth Que.....

5.0 Functional Description of the Character Generator.....

6.0 The Character Generator Algorithm and Control.....

7.0 Detailed Description of the Character Generator Cards.....

7.1 195221 Line Generator Character Multiplier.....
7.2 195222 Character Memory.....
7.3 195223 Character Control.....

8.0 Alignment and Tuning of the Line Generator.....8-1

8.1 Start Up Procedure.....8-1
8.2 X-Y Analog Card (195212) Slot 33.....8-2

8.2.1 Zeroing of the XY Integrator Output Offset.....8-2
Voltage
8.2.2 X and Y DAC Output Offset Voltage and.....8-3
Integrator Input Current Adjust

8.3 DISPLAY Phase Adjustment.....8-6
8.4 Further Adjustments on the X-Y ANALOG Card.....8-6
Compensation For Extend Integrator and X-Y
Integrator Switching Noise

8.4.1 Null Extend Integrator Switching Noise.....8-7
8.4.2 Null X-Y Integrator Switching Noise.....8-7

8.5 Scope Driver (195211) Endpoint-Match.....8-9
8.6 Scope Driver (195211) No Contrast.....8-11

8.6.1 Display Controls and Scope Driver 8-11
Potentiometers
8.6.2 No Contrast - Equal Vector Brightness.....8-13
Adjustment

8.7 Scope Driver (195211) Contrast Adjustments.....8-15

8.7.1 Video Delay.....8-15
8.7.2 Video DAC Glitch Attenuation.....8-16

9.0 Troubleshooting the Line Generator.....

10.0 Troubleshooting the Character Generator.....

1.0 INTRODUCTION

This volume of the PICTURE SYSTEM 2 Maintenance Manual covers the Line Generator (L.G.) and the Character Generator (C.G.) which are sometimes collectively referred to in other literature covering PICTURE SYSTEM 2 as the PICTURE GENERATOR.

The L.G. and C.G. are special purpose digital-analog machines which decode the instructions that command a vector to be drawn on the display and perform the necessary computation on the associated data to produce the X and Y deflection voltages, the Z axis intensity signals (incorrectly called video in some of the literature) and the CRT beam unblank voltage. As PICTURE SYSTEM 2 is also capable of driving beam penetration color displays the L.G. also decodes color instructions and provides five color select signals.

In order to repair and maintain the L.G. and C.G. the reader must be thoroughly familiar with the theory of operation of both machines which is to say that he must understand the computation that is performed. Experience has shown that the final display provides the troubleshooter with the information required to quickly isolate the problems in the vast majority of the cases encountered if he is thoroughly familiar with the computation that is performed. This manual is written and organized with that goal in mind.

The L.G. is the primary machine. The C.G. is a machine that provides particular inputs to the L.G.; those which describe vectors whose final displayed image takes the form of alpha numeric characters.

This manual first defines the equations solved by the L.G., Sections 2.1 and 2.2, then traces the flow of the solution through the machine, Section 2.3. The computational algorithm and the implementation of the control logic is then described in Section 3.0. The detailed description of the implementation of the arithmetic and control functions is covered by a detailed description of each L.G. circuit card, Section 4.0. A similar treatment is given to the C.G., Section 5.0 describes the data manipulation used to generate alpha numeric characters and Section 6.0 describes the C.G. algorithm and the control logic that implements it. A detailed description of the C.G. circuit card logic is presented in Section 7.0

Section 8.0 presents the procedure to be followed to align and tune the analog circuits that drive the display. General troubleshooting advice is given for the L.G. in Section 9.0 and the C.G. in Section 10.0. The contents of each PROM used in the L.G. and C.G. is listed in Appendix A.

2.0 THE LINE GENERATOR COMPUTATION

The function of the Line Generator is to produce the X and Y deflection voltages, the electron beam unblank input and the Z axis intensity (video) input for the display. The input data which are presented to the Line Generator are:

-- Xn, Yn - The display plane coordinates of the NEW end point specified by a DRAW or MOVE command.

-- Zn - The intensity desired at the NEW end point coordinate.

-- Command Data - This data defines the characteristics of the beam deflection that is to exist as the beam is deflected to the NEW point. i.e. A MOVE, a DRAW, a dashed line, a blinking line, etc.

Operating on these data the Line Generator performs the operation of generating the deflection and beam intensity information required as the beam is moved from the present position or OLD end point, (Xo, Yo, Zo) to the desired position or NEW end point, (Xn, Yn, Zn).

2.1 The Mathematics of Line Generation

The mathematical relationships which are used to define the straight line between (Xn, Yn) and (Xo, Yo) which is to be

"drawn" on the display are shown in Figure 2-1. Physically the Line Generator performs the task of generating the X and the Y deflection voltages Vx and Vy that are applied to the display to move the beam from Xo, Yo to Xn, Yn.

A line that is to be drawn between the points (Xo, Yo) and (Xn, Yn) on the display is mathematically represented as the hypothenuse of a right triangel. The familiar relationship that exist in a right triangle are shown in Figure 2.1

$$\text{length} = \sqrt{(\Delta X)^2 + (\Delta Y)^2}$$

where: $\Delta X = Xn - Xo$
$\Delta Y = Yn - Yo$

The cosine of the angle Θ:

$$\text{COS } \Theta = \frac{\Delta X}{\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

The sine of the angle Θ:

$$\text{SINE } \Theta = \frac{\Delta Y}{\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

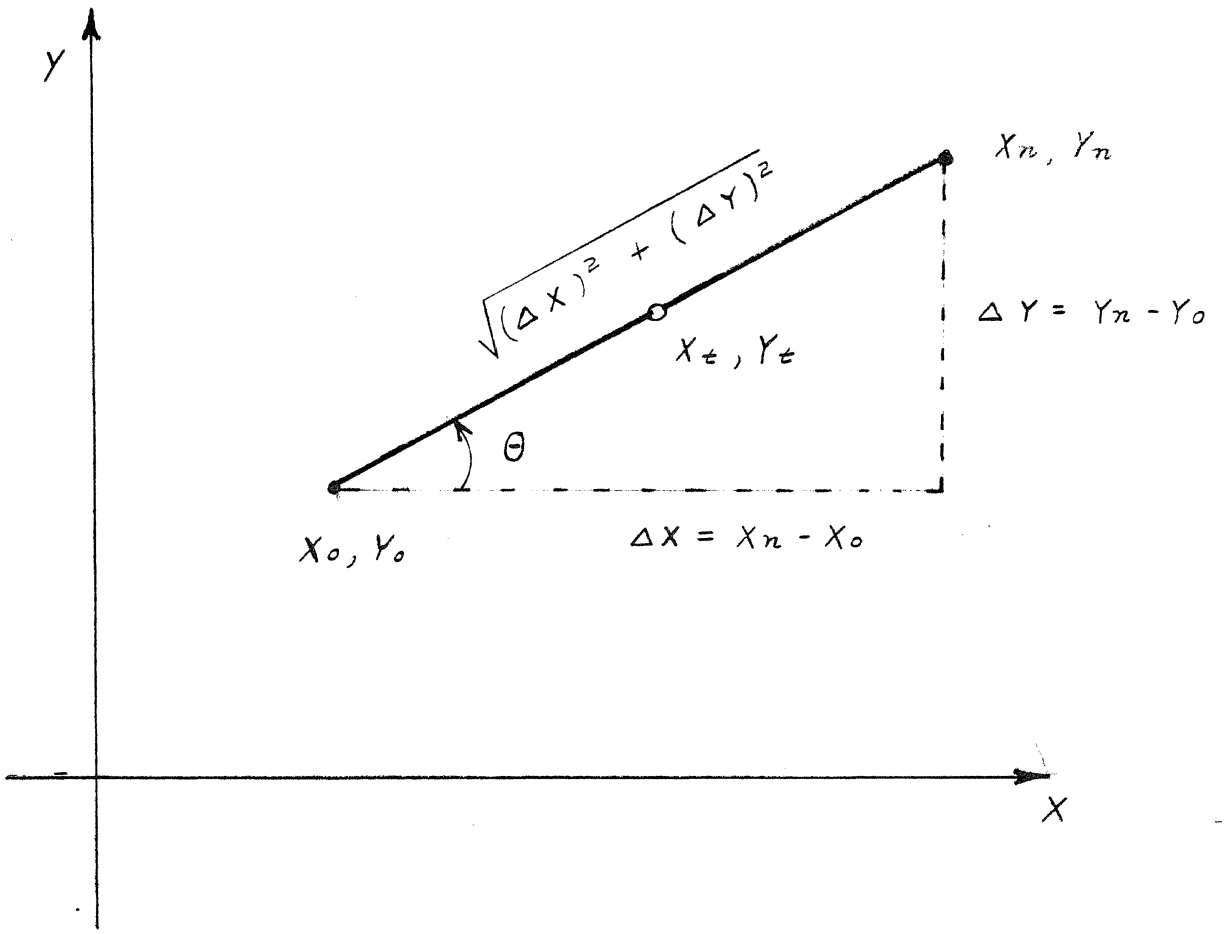


Figure 2.1
RELATIONSHIPS IN A RIGHT TRIANGLE

The locus of a point (X_t, Y_t) on the line is given by:

$$X_t = Xo + t \text{ COS } \Theta$$
$$Y_t = Yo + t \text{ SIN } \Theta$$

where t is a parameter that has the following property:

$$0 \leq t \leq \sqrt{(\Delta X)^2 + (\Delta Y)^2}$$

If the parameter t is a function of time then the locus of the point X_t, Y_t is also a function of time.

2.2 Converting to the Time Domain

Figure 2.2 is a simplified functional diagram showing how the Line Generator generates the time varying X and Y deflection voltages. The digitally computed COS Θ and SIN Θ quantities are applied to D/A converters and the two's compliment of the length of the line to be drawn is the initial count loaded into the Integration Time Counter. The time counter output signal INTEGRATE goes high closing the X and Y switches and the COS Θ and SIN Θ quantities are applied to the X and Y integrators. The output voltages of the integrators are proportional to the respective cosine and sine. The number of CLK periods that COS Θ and SIN Θ are applied to the integrators is controlled by the compliment of the line length that was initially loaded into the Integration Time Counter. When the

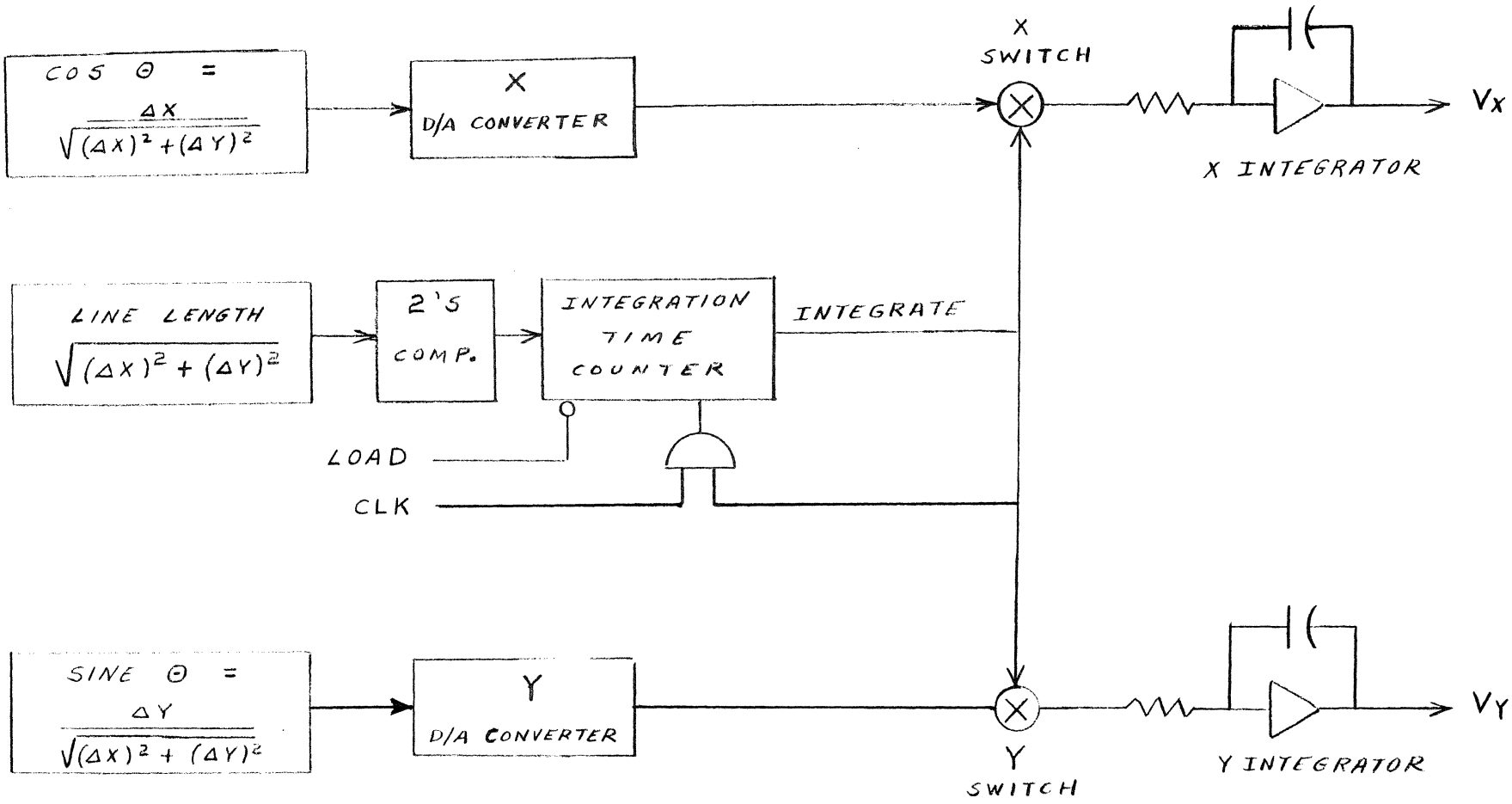


Figure 2.2

THE TIME DOMAIN MULTIPLICATIONS

time counter has counted up to "all ones" the signal INTEGRATE goes low causing the X and Y switch to open. Hence the X and Y deflection voltages no longer change. This time controlled integration process performs the time domain multiplications t COS Θ and t SINE Θ required to produce the X and Y deflection voltages.

2.3 Computational Tasks

In order to generate the time domain deflection voltages the L.G. computes the following quantities in the order as listed:

a) Computes the X and Y differences, ΔX and ΔY, between the NEW and OLD endpoints.

$$\Delta X = Xn - Xo$$
$$\Delta Y = Yn - Yo$$

b) Determines which difference has the larger numerical value.

ΔX or ΔY

ie. $\Delta X \geq \Delta Y$

c) Block normalizes ΔX and ΔY by left shifting them the same number of places. The number of left shifts, N, is determined by the number of leading zeros.

if $\Delta X \geq \Delta Y$ $N = Nx$
if $\Delta X < \Delta Y$ $N = Ny$

d) Squares the normalized deltas and computes their sum.

$$(N\Delta X)^2 + (N\Delta Y)^2$$

e) Calculates the inverse of the square root of the sum of the normalized squares.

$$\frac{1}{\sqrt{(N\Delta X)^2 + (N\Delta Y)^2}} = \frac{1}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

f) Calculates the square root of the sum of the normalized squares.

$$\sqrt{(N\Delta X)^2 + (N\Delta Y)} = N\sqrt{(\Delta X)^2 + (\Delta Y)^2}$$

g) Calculates the sine and cosive of the angle Θ. i.e. The direction cosine and sine of the line (Xn - Xo), (Yn - Yo).

$$(N\Delta X)\left\langle \frac{1}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}} \right\rangle = \frac{\Delta X}{\sqrt{(\Delta X)^2 + (\Delta Y)^2}} = \text{COS } \Theta$$

$$(N\Delta Y)\left\langle \frac{1}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}} \right\rangle = \frac{\Delta Y}{\sqrt{(\Delta X)^2 + (\Delta Y)^2}} = \text{SIN } \Theta$$

h) Calculates the length of the line by un-normalizing the quantity $N\sqrt{(\Delta X)^2 + (\Delta Y)^2}$.

i) Generates the X and Y deflection voltages by performing a integration of COSINE Θ and SINE Θ

$$V_x \triangleq X_t = Xo + t \text{ COS } \Theta$$
$$V_y \triangleq Y_t = Yo + t \text{ SIN } \Theta$$

2.4 The Line Generator Data Flow

A block diagram illustrating the data flow through the Line Generator for execution of the DRAW and MOVE commands is shown in Figure 2.3. This diagram is more detailed version of the 195210-900 drawing. The dashed outline defines the card on which the functions are implemented and the circled number is the backpanel slot in which the card resides.

2.4.1 The Input FIFO'S

The Line Generator is interfaced to the PSBUS by the 195219 LG FIFO PORT card. The card contains two four word by 16 bit register files, the LEFT FIFO and the RIGHT FIFO. These register files are implemented by 74LS60 4 x 4 tri-state memory chips. The FIFO's are loaded with input data from the PSBUS. In a standard Picture System configuration the source of the data is the PSMEM although it is possible to source the L.G. input data from any device capable of outputting to the PSBUS. The LEFT FIFO is loaded with X data and the RIGHT with Y data and their respective onputs are named XCBUS and YCBUS. These output busses can also be

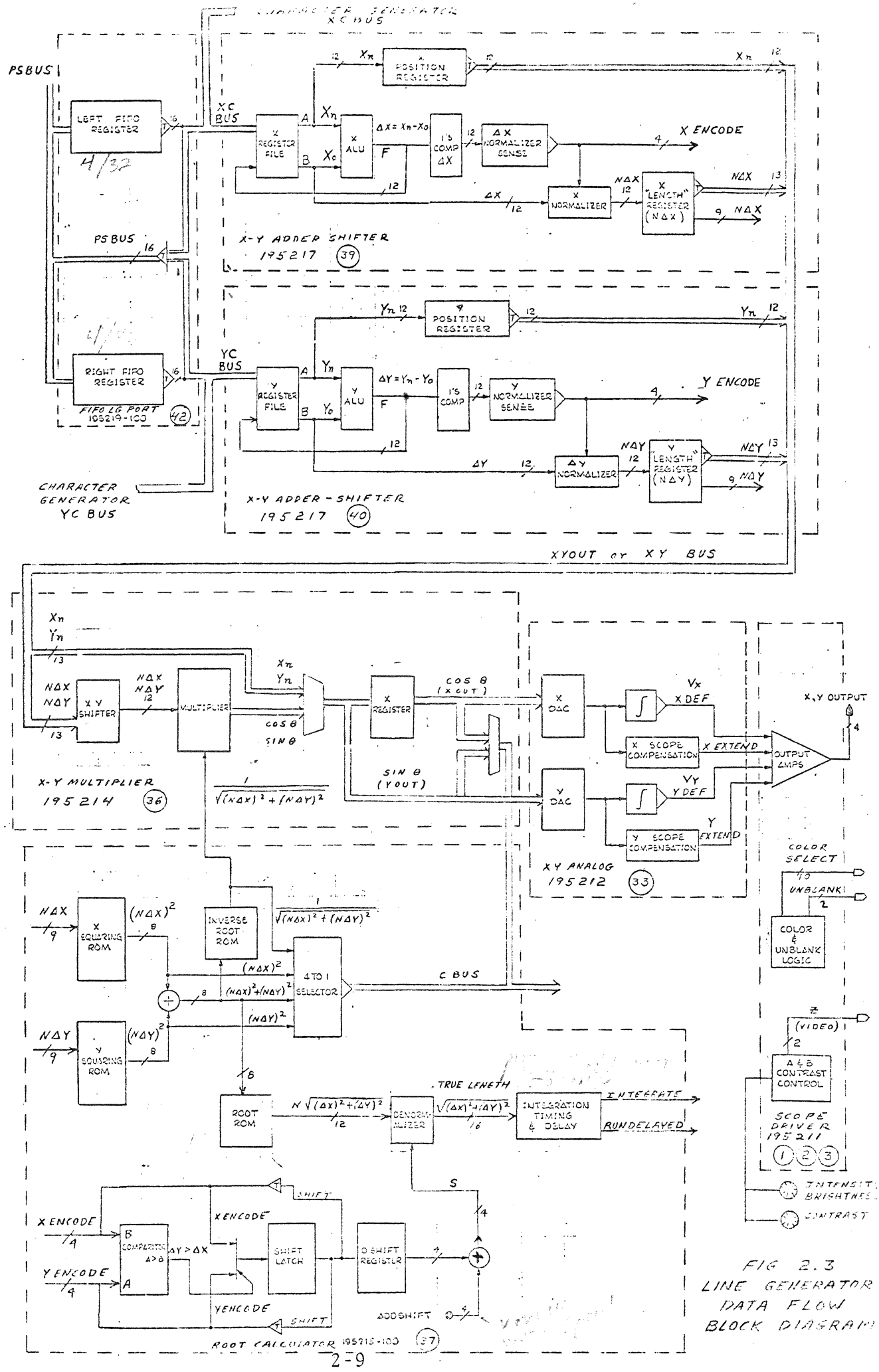


FIG 2.3
LINE GENERATOR
DATA FLOW
BLOCK DIAGRAM

multiplexed back onto the PSBUS for maintenance purposes. The LG FIFO PORT card also contains the logic required to decode its own address and to keep track of the status of the FIFO's. This later logic consists of read and write address counters and a status ROM. The status ROM indicates if the FIFO's are full and prohibits the acceptance of new input data if the full condition exists.

The data stored in the FIFO's are the Line Generator and Character Generator data and command words. These 16 bit words are received from the PSBUS. The Line Generator removes two 16 bit words from the FIFO and inputs them as a single 32 bit word. The four 32 bit words are:

MOVE or DRAW
STATUS
REFRESH CONTROL
CHARACTER

The assignment of the bits of the words to the XCBUS and YCBUS lines are shown in tables 2-1 and 2-2 for the four words. A more detailed definition of the word bits is presented in Section 2.4.3 of the PICTURE SYSTEM 2/PDP-11 REFERENCE MANUAL. Note that the bits which contain the code for identifying commands are bits 15, 14 and 7 of word 1 and have the following XCBUS assignments:

	BIT	MOVE OR DRAW	STATUS
XCBUS	15	=1, Specifies Move or Draw	=0, These bits along with bit 7
	14	0= MOVE, 1= DRAW	=0, Specify a STATUS command
WORD 1	13	Two MSB	=1 Blink
	12	Of Z Z(5) and Z(4)	=1 Display dashed line end point, =0 do not
	11	The 12 bits	TEXTURE 100 long dash lines
	10	Specifying the	000 Solid 101 long short 110 long-short-short
	9	Absolute value	001 dot 111 short
	8	Of the X coordinate	=1 RELATIVE
	7	Of the line end point	=1 STATUS command
	6		PFORM CCHAR perform ASCII 26, 27
	5		CG RESET reset character generator
	4		COLOR 101 Orange 000 Mono
	3		011 Red 110 Yellow 001 Mono
	2		100 Red-Or 111 Green 010 Mono
	1		INTENSITY 01 2X Norm (Grn,Yel) 10 4X Norm (Or,Red-Or)
	0		00 Normal 11 8X Norm (Red)

YCBUS	15	Lower 4 bits	SCOPE Scope 6 5
	14	Of Z	SELECT Scope 5 4
WORD 2	13	Z(3), Z(2), Z(1), Z(0)	Scope 4 3
	12		Scope 3 2
	11	The 12 bits	Scope 2 1
	10	Specifying the	Scope 1 0
	9	Absolute value	UNUSED
	8	Of the Y coordinate	
	7	Of the line end point	
	6		
	5		
	4		
	3		
	2		
	1		
	0		Char Intensity =1 intensity from Font Par.

TABLE 2-1: MOVE or DRAW and STATUS word: Bit - Bus Assignment

	BIT	REFRESH CONTROL	CHARACTER COMMAND
XCBUS	15	=0 Along with bit 7	=0 Character Command
	14	=1 Specify REFRESH CONTROL	CHARACTER CODE
WORD 1	13	=1 HALT end of refresh seg.	2nd character
	12	SEGMENT 01 Jump	
	11	00 no function 10 Name 11 Name Blanked	
	10	LIGHT PEN	
	9		
	8		
	7	=1 REFRESH CONTROL	=0 CHARACTER COMMAND
	6	NOT	CHARACTER CODE
	5	USED	
	4		1st character
	3		
	2		
	1		
	0		

YCBUS	15	Eight MSB's of SEGMENT	or'ed with Character Style LSB
	14	Name	CHARACTER CODE
WORD 2	13	(May be masked	
	12	off as a byte	
	11	by RFSR neg.)	4th character
	10		
	9		
	8		
	7	Eight LSB's of SEGMENT	or'ed with Character Style LSB
	6	Name	
	5	(May be masked	
	4	off as a byte	
	3	by RFSR neg.)	3rd character
	2		
	1		
	0		

TABLE 2-2: REFRESH CONTROL and CHARACTER COMMAND Bit - Bus Assignment

	XCBUS(15)	XCBUS(14)	XCBUS(7)
MOVE	1	0	X
DRAW	1	1	X
STATUS	0	0	1
REFRESH	0	1	1
CHARACTER	0	X	0

2.4.2 X and Y Shifter Adders

The XY coordinate data of the new end point taken from the L.G. INPUT FIFO's is processed by the X, Y Adder Shifter Cards, 195217. These cards calculate the difference between the new and old end point coordinates:

$$\Delta X = Xn - Xo$$

$$\Delta Y = Yn - Yo$$

and block normalize these delta quantities to produce NΔX and NΔY. The old and new coordinate end points are stored in the X and Y register files. These files store eight words, location $\emptyset$ contains the new end point and location 1 the old end point. These are respectively the quantities XYIN and XY as shown at the lower left of the ARITHMETIC CONTROL ARITH-SHIFTER CONTROL LOGIC, 195216-950 drawing. The X and Y ALU's are constructed of 74S181's. They are capable of performing four operations as used on the 195217 card:

Pass the A input	F ← A
Add	F ← A + B

Subtract	F ← A - B
Clear	F ← Ø

The ALU performs the subtraction required to obtain ΔX = Xn - Xo. The 1's COMPLIMENTER is used to compliment a negative ΔX or ΔY as this is required by the block normalization method used. The X and Y NORMALIZER SENSE examines the Δ quantity to determine the number of leading zeros or the number of left shifts required to block normalize the quantity. These numbers are X ENCODE and Y ENCODE which are compared on the 195215, ROOT CALCULATOR card. The smaller of these two values, which is obtained from the larger of ΔX and ΔY, is used to normalize both ΔX and ΔY. The shift code N is saved for use in a subsequent denormalizing operation. The normalized Δ quantities, NΔX and NΔY are stored in the X and Y LENGTH REGISTER's. These registers have a dual output capability. Thirteen bits are available to the tri-state XYOUT bus. Nine bits, totem pole output, are sent to the LINE GENERATOR ROOT CALCULATOR, 195215 card. This is the normal data flow path which is used during execution of a DRAW instruction. A thirteen bit normalized ΔX and ΔY are sent to the LINE GENERATOR XY MULTIPLIER, 195214, card on the XYOUT bus and a nine bit normalized ΔX and ΔY are sent to the LINE GENERATOR ROOT CALCULATOR, 195215, card.

When a MOVE instruction is executed the new end point coordinate data Xn and Yn are used in the time domain integration instead of the sine and cosine quantities:

$$\text{COS } \Theta = \frac{\Delta X}{\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

$$\text{SINE } \Theta = \frac{\Delta Y}{\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

The delta quantities are still used to control the integration time however the time is controlled by $1/2\ \sqrt{(\Delta X)^2 + (\Delta Y)^2}$ thus making the time for a MOVE one half that of a DRAW. This supplying of Xn and Yn to the 195214 for a MOVE is done via the X and the Y POSITION REGISTERS whose tri-state output is switched onto the tri-state XYOUT bus in place of the normalized delta quantities. For a MOVE the nine bit normalized delta quantities are supplied to the ROOT CALCULATOR just as is done for a DRAW.

2.4.3 XY Multiplier

The sine and cosine functions:

$$\text{COS } \Theta = \frac{N\Delta X}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

$$\text{SINE } \Theta = \frac{N\Delta Y}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

are calculated on the 195214 XY MULTIPLIER card. The normalized delta quantities $N\Delta X$ and $N\Delta Y$ are input from the XYOUT

Bus. The normalized inverse square root:

$$\frac{1}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

is input from the 195215, ROOT CALCULATOR. The calculated cosine and sine functions are respectively presented to the X and Y D/A converters on the 195212 X Y ANALOG CARD.

COSINE Θ is stored in the X REGISTER until SINE Θ is calculated by the multiplier so that both cosine and sine can be simultaneously input to the X Y analog card. The 2 to 1 multiplexer between the MULTIPLIER and the output is used to select the MULTIPLIER output COSINE Θ and SINE Θ during a DRAW or to select Xn and Yn during a MOVE. The outputs of the XY MULTIPLIER can also be selected onto the C BUS for maintenance purposes.

The X Y SHIFTER at the input to the MULTIPLIER is used to right shift the ΔX and ΔY quantities to obtain a smaller value of cosine and sine when the integration rate is slowed for color displays and for the drawing of characters.

2.4.4 The Square Root Calculations

The square root of the sum of the squares of ΔX and ΔY is used to calculate the COSINE Θ, SINE Θ and to determine the integration time for the DRAW and MOVE commands. These

quantities are calculated on the 195215, ROOT CALCULATOR card.

The N(ΔX) and N(ΔY) quantities are squared by a ROM look up process and added. The square root and the inverse square root of the sum are also found by a ROM look up process. The square root and its inverse are normalized versions because the original delta quantities were normalized. The demormalized square root, which is the true length of the line, is used to control the integration time. The denormalization compares the X ENCODE and Y ENCODE values and uses the smaller of the two to determine the number of right shifts of the square root required to denormalize. The sixteen bit true value of the square root is used to determine the integration time for the DRAW instruction being executed.

2.4.5 The X - Y Analog Function

The COSINE Θ and SINE Θ values, or Xn and Yn in the case of a MOVE, are applied to D to A converters whose analog output supplies the input source current for a standard R-C integrator circuits.

These functions are performed on the 195212 X-Y ANALOG CARDS. Supplementary circuits on this card modify the current supplied to the integrators to make the adjustments required to handle color displays and to provide the correct amount of beam overshoot to insure that the specified end point match

can be achieved dispite differences in display amplifier delay time characteristics.

2.4.6 Scope Drive, Contrast, Color Drive and Unblank

The 195211 Scope Driver contains the X and Y deflection amplifiers, the variable tapped delay lines along with switched UNBLANK amplifiers, these switches generate BLANK and UNBLANK levels for the various blink and dash modes of line drawing, the contrast control circuitry and the color decode and delay circuitry for driving color displays. Each scope driver card contains dual sets of these circuits and is thus capable of driving two seperate displays.

3.0 THE LINE GENERATOR CONTROL STATE MACHINE

3.1 The Four State Machine Controllers

There are four PROM implemented state machines employed to generate the signals which control the data flow in the Line Generator. A fifth PROM implemented state machine is employed to generate the control signals for the Character Generator.

The four L.G. state machine controllers and the C.G. state machine controller are shown in Figure 3.1. The four controllers operate in a pipe line fashion. The PROM address which selects the initial state of a sequence of states for the Arithmetic Controller, Multiplier Controller and the Output Controller is obtained from the preceeding controller. The L.G. Input Controller, shown at the left, receives the address bits which determine its initial state from the XCBUS via the A input of the 2 to 1 multiplexer. These XCBUS bits are those which define the input command to the L.G., XCBUS 15, 14 and 7. The succeeding states that the Input Controller advances to are determined by address bits stored in its own PROM, INPUTCTLPROM which are fed back to the PROM address lines via the B input of the 2 to 1 multiplexer. The change of state occurs when the INPUT STATE CHANGE logic causes the *CHANGEDATACTL to go low enabling the INPUTCTLPROM outputs to be loaded into the INPUTCTLREG by the clock. The INPUTCTLPROM output bits OPN are passed forward to the Arithmetic Controller

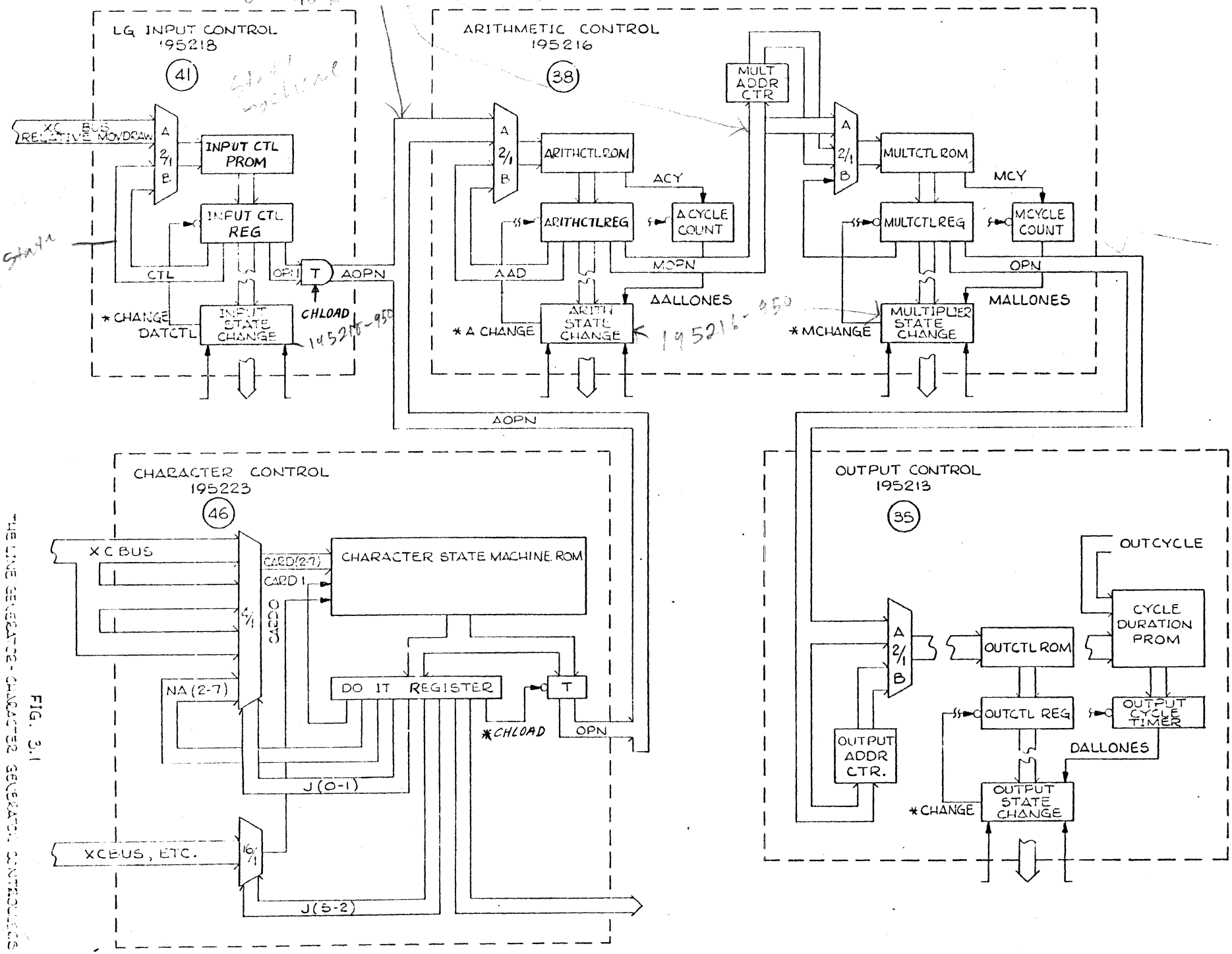


FIG. 3.1 THE LINE GENERATOR - CHARACTER GENERATOR CONTROLLERS

via the tri-state bus AOPN. These bits are applied to the ARITHCTLROM address by input A of a 2 to 1 multiplexer. The Character Generator is also interfaced to the L.G. controllers by the AOPN bus. The signal which determines if the address passed to the Arithmetic Controller to select its initial states comes from the Input Controller or the Character Controller is CHLOAD. The outputs of the ARITHCTLROM are loaded into the ARITHCTLREG by the clock when the ARITH STATE CHANGE logic makes the register enable, *ACHANGE, low. The Arithmetic Controller contains the two bit ACYCLECOUNT counter whose initial count is determined by the ACY bits of ARITHCTLROM. This counter is counted by the clock and when it reaches the binary count 1111 the signal AALLONES goes high enabling the ARITH STATE CHANGE logic to generate *ACHANGE low. Thus this counter determines the number of clock periods that the Arithmetic Controller remains in a particular state. The ARITHCTLREG output bits MOPN are passed forward to address the MULTCTLROM via input A of the Multiplier Controllers 2 to 1 multiplexer. The operation of the Multiplier Controller is similar to that of the Arithmetic Controller. The MULTCTLREG, the MULTIPLIER STATE CHANGE logic, the MCYCLE COUNTER and the signal *MCHANGE play roles analogous to the Arithmetic Controller ARITHCTLREG, ARITH STATE CHANGE logic, ACYCLE COUNTER and the signal *ACHANGE. The Multiplier Controller also contains the four bit MULT ADR CTR whose upper two bits are preloaded with two of the four bits passed over by the Arithmetic Controller. Succeeding states of the Multiplier Controller

are generated by applying the MULT ADR CTR outputs to the address inputs of the MULTCTLROM via the B input of the 2 to 1 multiplexer. The MULT ADR CTR is counted by the clock and as might be expected sequential states of the Multiplier Controller have consecutive state numbers. The OPN output bits of the MULTCTLREG are passed to the Output Controller and address the OUTCTLROM via the A input of the 2 to 1 multiplexer. The Output Controller is similar in structure to the Multiplier Controller. It has an additional feature, the CYCLE DURATION PROM, which provides a large range of initial counts for loading into the OUTPUT CYCLE TIMER. This PROM provides a variety of clock period durations for various states of the Output Control state machine.

3.2 The Input Control State Machine (195218)

The following figures, drawings and tables are necessary to the discussion of the Input Control State Machine that follows:

a. Figure 3.2 Input Control State Machine - Block Diagram
b. Figure 3.3 Input Control State Change Logic
c. 195218-600 L.G. Input Control Logic
d. 195218-950 L.G. Input Control State Diagram
e. Figure 3.4 Detailed L.G. Input Control State Diagram
f. Table 3.1 INPUTCTLPROM and INPUTCTLREG Address - Data Table

Figure 3.2 is a simplified block diagram of the L.G. Input Control Logic shown on sheets 3 and 4 of the 195218 Logic Diagram. The control logic is an implementation of a finite state machine. The data defining the control signals for each state are stored in a 32 word by 16 bit PROM, INPUTCTL-PROM. The output of INPUTCTLPROM are loaded into the INPUT-CTLREG by the clock when *CHANGEDATA is low. The state diagram for the Input Control is shown on the 195218-950 drawing. There are eight different sequences of states that the machine may dispatch to. The number of the first state and its INPUTCTLPROM address, the name of the sequence of states, and the operation the L.G. is to perform are:

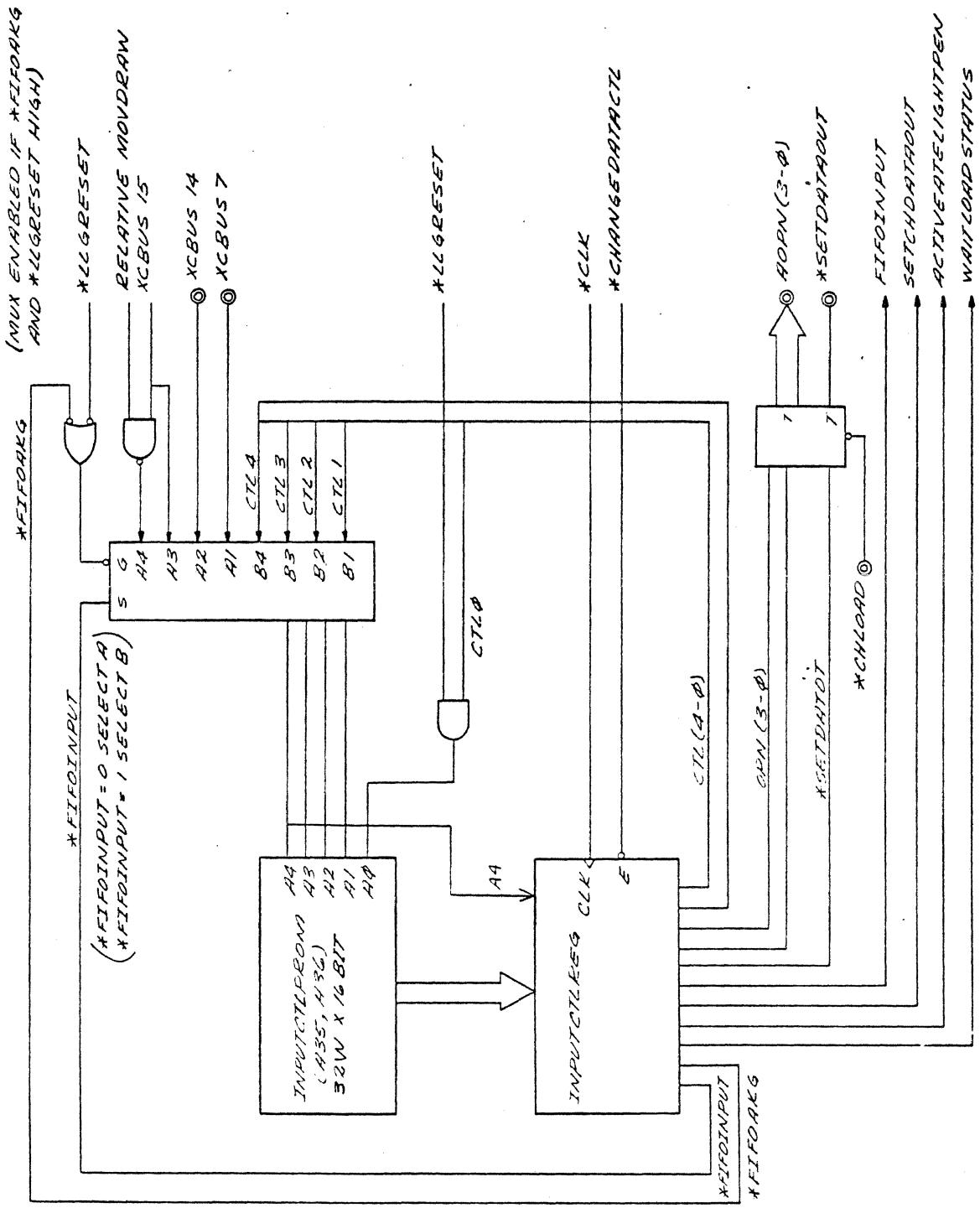


FIGURE 3.2
INPUT CONTROL STATE MACHINE BLOCK DIAGRAM
(REF 195128 SHEET 3+4)

STATE NUMBER and INPUTCTLPROM Address		DISPATH SEQUENCE	OPERATION
OCTAL	BINARY	NAME	
ØØ	ØØØØØ	RESET	Line Generator Reset
1Ø	Ø1ØØØ	RMOV	Relative Move
12	Ø1Ø1Ø		
3Ø	11ØØØ	MOV	Move
32	11Ø1Ø		
14	Ø11ØØ	RDRAW	Relative Draw
16	Ø111Ø		
34	111ØØ	DRAW	Draw
36	1111Ø		
22	1ØØ1Ø	STATUS	Status Command
26	1Ø11Ø	LIGHTPEN/ STATUS	Lightpen Command
20	1ØØØØ	CHARACTER	Character Command
24	1Ø1ØØ		

The numbers in the upper left hand corner of each state block of the diagram is the state number in octal and it is also the IPUTCTLPROM address of the control data for that state. The control signals that are activated during the state are contained in the state box. ie. State 22, *SETDATOT is low during this state. In the state boxes for the second state of each sequence, states 11,31,15 etc., the statement SEND RMOV, SEND MOV, SEND RDRAW etc., appears. These statements refer to the OPN (3-Ø) bits that are sent forward by the Input Controller to address the Arithmetic Controller. These bits are applied to a 2 to 1 multiplexer to the upper four bits of the ARITHCTLROM. These codes are:

STATE	NAME	OPN(3-0) CODE	ARITHCTLROM ADDRESS
11	RMOV	0100	1Ø
31	MOV	1111	36
15	RDRAW	0110	14
35	DRAW	0010	04
23	STATUS	0001	02
27	LIGHTPEN	1100	3Ø

Figure 3.4 contains a more detailed state diagram. A brief explanation of the functions performed in each state appears adjacent to the state box.

The four upper address inputs to the INPUTCTLPROM, see Figure 3.2, are selected by a 2 to 1 multiplexer. Those selected by the A input, the situation when *FIFOINPUT is low, are XCBUS 15 logically NANDED with RELATIVEMOVDRAW, XCBUS 15, XCBUS 14 and XCBUS 7. The XCBUS 15, 14 and 7 inputs are those which specify the MOVE, DRAW, STATUS, REFRESH and CHARACTER commands as presented in Section 2.3.1 above. RELATIVEMOVDRAW comes from the 195216 Arithmetic Control Card, slot 38. It is the signal XCBUS 8 which is stored in a flip flop, see sheet 4 of the 195216 logic, and is actually bit 8 of the STATUS command. The address bits selected by the B input, when *FIFOINPUT is high, are the CTL4 through CTL1 bits of the INPUTCTLREG which were loaded into this register from the INPUTCTLPROM. The least bit of the address is the logical AND of *LLGRESET, the Line Generator reset, and CTLØ. It will be low when the Line Generator is reset or when CTLØ is low. The multiplexer select

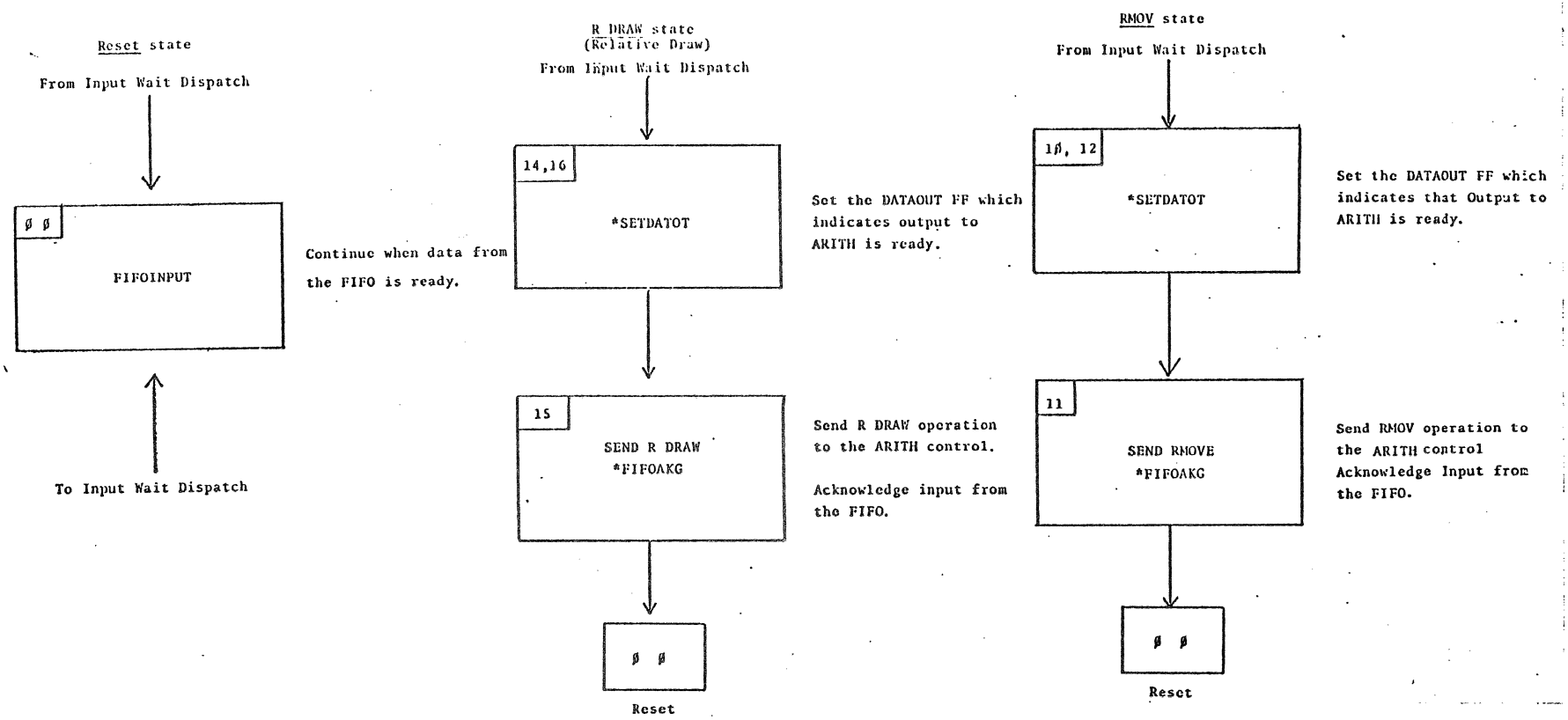


Figure 3.4

DETAILED STATE DIAGRAM INPUT CONTROLLER

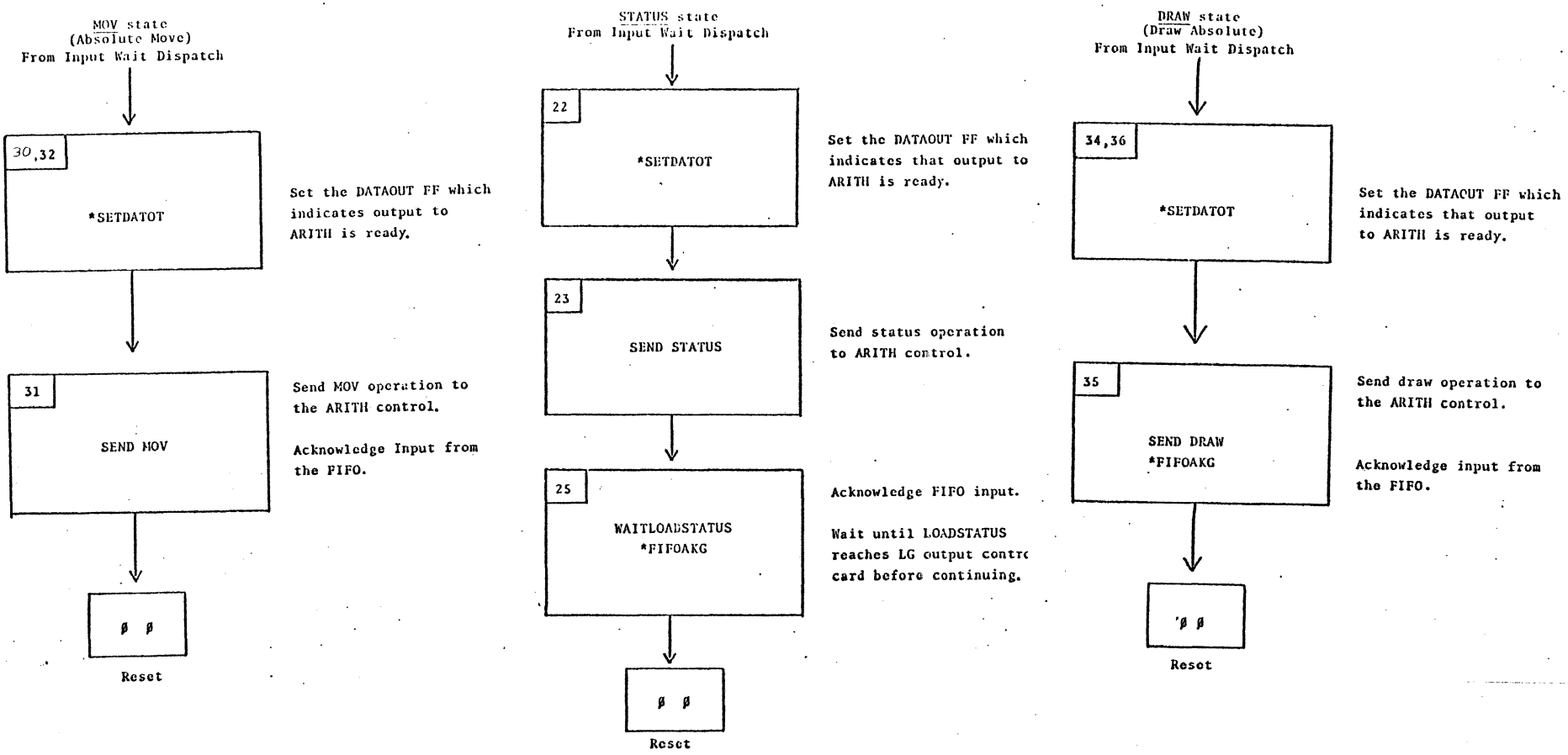


Figure 3.4

DETAILED STATE DIAGRAM INPUT CONTROLLER

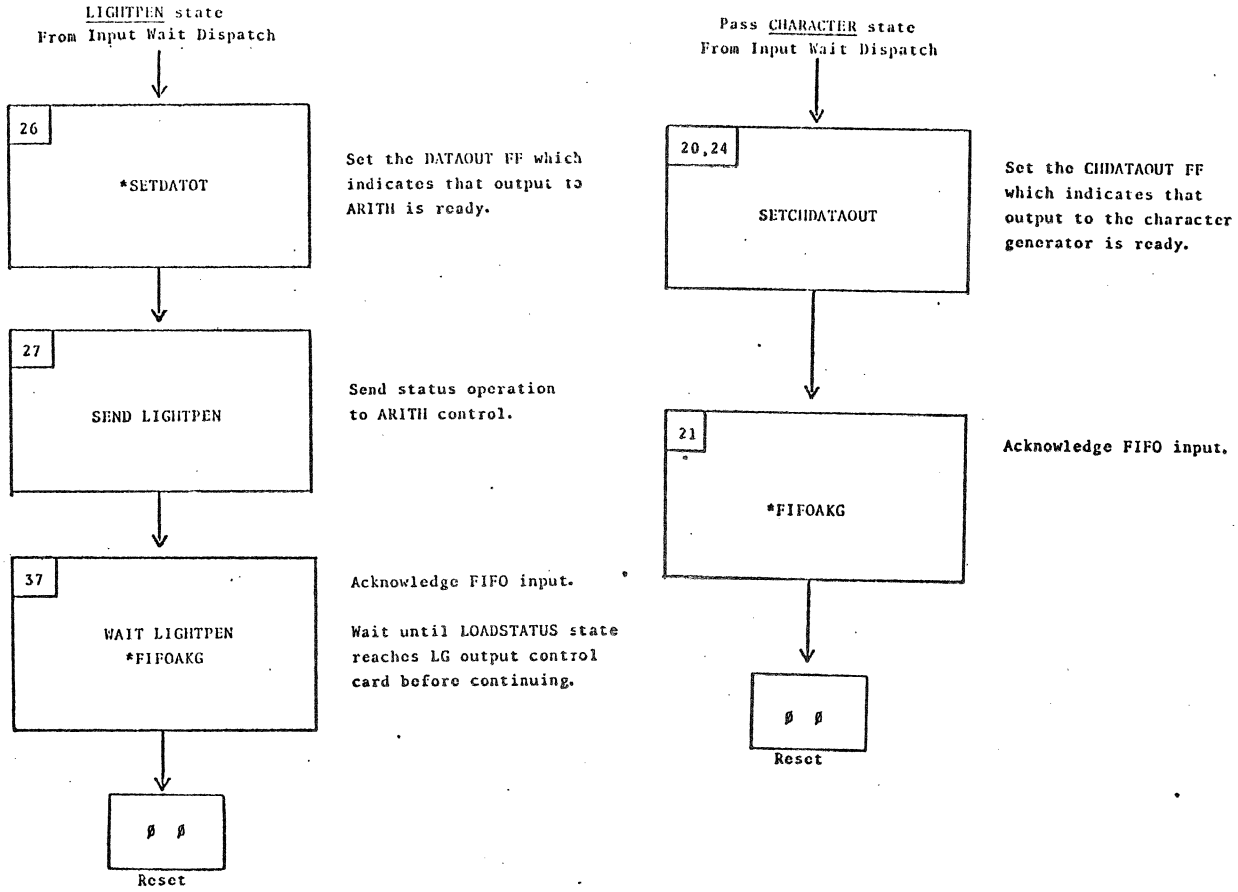


Figure 3.4
DETAILED STATE DIAGRAM INPUT CONTROLLER

is driven by *FIFOINPUT which is also a INPUTCTLPROM bit. It is low, select A condition, only for state Ø, Ø of the controller. Thus when the controller is in state Ø, Ø, the contents of INPUTCTLREG are state Ø, Ø, the controller is awaiting a input from the FIFO. The multiplexer enable input is pulled high, its outputs therefore forced to the low state, when *FIFOAKG or *LLGRESET, the Line Generator reset, are low. *FIFOAKG is an INPUTCTLPROM output. It and CTLØ are both low for the last states of each sequence of states and force the controller to return to state Ø, Ø. Hence both these bits are low for states 11, 31, 15, 35, 25, 37 and 21.

The INPUTCTLPROM data is clocked into the INPUTCTLREG only when the register enables *CHANGEDATACTL is low. *CHANGEDATACTL is generated by the logic shown in Figure 3.3, Input Control State Change Logic. This is a simplified version of sheet 4 of the 195218 logic drawing and also shows the handshaking logic between the Input Controller and the FIFO.

Gate 13.12 is an "OR" gate which allows *CHANGEDATACTL to go low when the Line Generator is reset, *LLGRESET goes low, or SETCHDATAOUT, one of the outputs of INPUTCTLREG, is high or the output of gate 60.9, is low. The operation of gate 60.9 is best visualized as performing the function of preventing *CHANGEDATACTL from going low by virtue of having one of its inputs low. Generally all the inputs to 60.9 with the exception of one will be high. This one low input keeps *CHANGEDATACTL high.

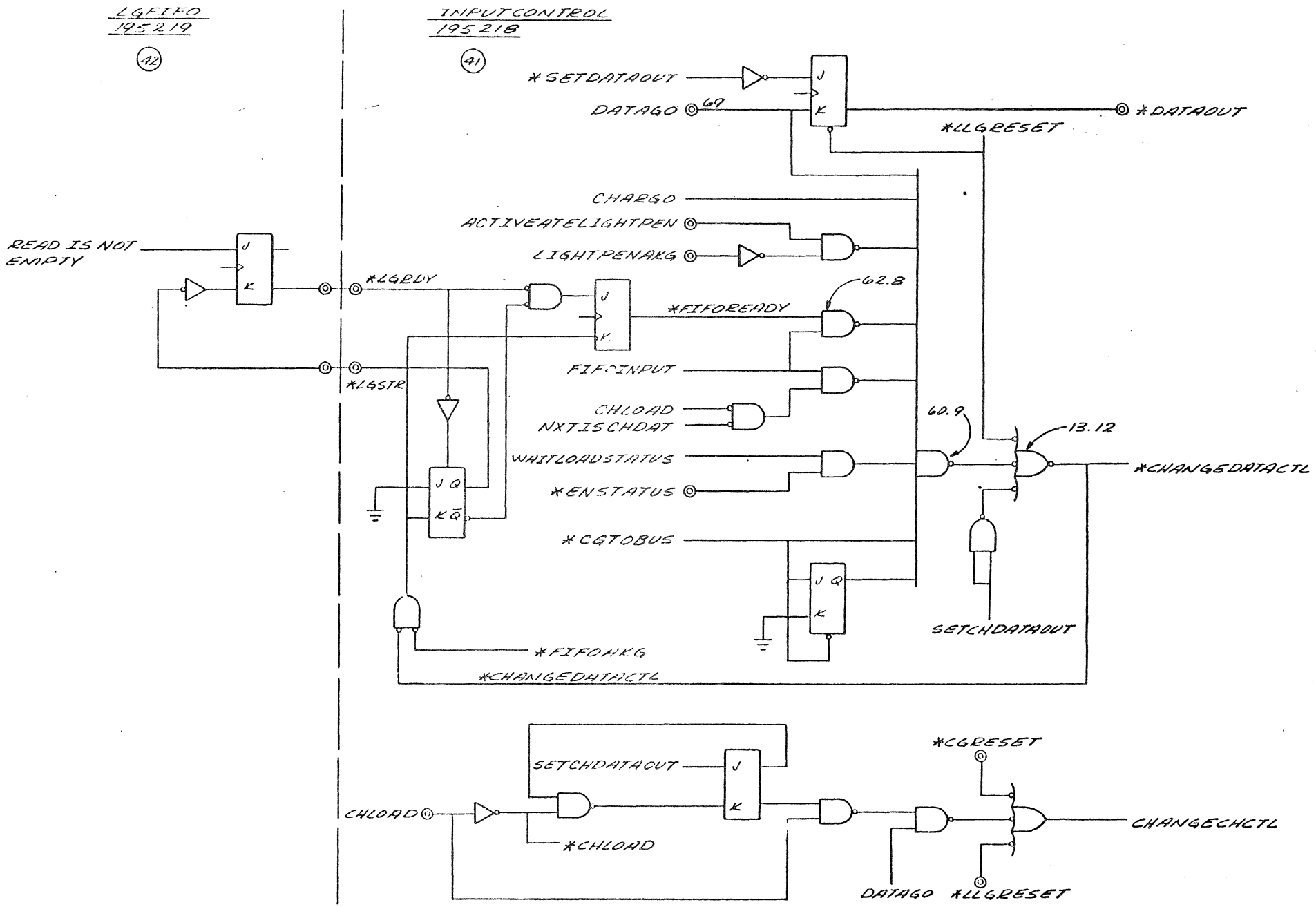


FIGURE 3.3
INPUT CONTROL STATE MACHINING-
STATE CHANGE LOGIC

Consider as an example the FIFO input sequence. The inputs to gate 62.8, *FIFOREADY and FIFOINPUT, are both high therefore its output is low and *CHANGEDATACTL is high. *FIFOREADY is high because the FIFO has not yet pulled *LGRDY low to signal that the FIFO has an input. FIFOINPUT is high because the Input Controller is in state ØØ awaiting a FIFO input. When the FIFO pulls *LGRDY low the *FIFOREADY flip flop is set low on the next clock pulse. The output of 62.8 goes high and *CHANGEDATACTL is pulled low. When the controller was in state Ø, Ø the contents of INPUTCTLREG were those of state Ø, Ø. *FIFOAKG was high enabling the multiplier and *FIFOINPUT was low causing the multiplexer to select the A inputs which are those instruction code signals, XCBUS 15, 14, 7, that specify the state that the Input Controller is to dispatch to. Thus the INPUTCTLPROM outputs were those of the state to be dispatched to. On the next clock pulse, when *CHANGEDATACTL is low, the output of INPUTCTLPROM are loaded into INPUTCTLREG putting the controller into its next state.

Table 3.1 shows the contents of INPUTCTLPORM and the subsequent contents of INPUTCTLREG for each state; PROM address. The left most two columns list the state and INPUTCTLPROM address, in octal and in binary. The PROM A35 and A36 columns list the outputs by the B number of the PROM outputs. The names associated with each B number are the corresponding INPUTCTLREG names as appear on the logic. Note that the CTL4 output of INPUTCTLREG is the previous highest order address into INPUT-

INPUT CONTROL STATES (OCTAL)	INPUTCTLPROM ADDRESS A4 A3 A2 A1 A0	CTL4 ≡ A4 CTL3 ≡ (B7, PROM A36)	PROM A35 B7 B6 B5 B4 B3 B2 B1 B0								PROM A36 B7 B6 B5 B4 B3 B2 B1 B0							
			* FIFOARG	ACTIVATELIGHTPEN	WHITLOADSTATUS	OPN3	OPN2	OPN1	OPN0		(B7)	SETCHDATHOUT	FIFOINPUT	* FIFOINPUT	* SETDATAOT	CTL2	CTL1	CTL0
0	0 0 0 0 0	0 0	0	1	0	0	0	0	0		0	0	1	0	1	0	0	0
1	0 0 0 0 1	0 0	0	1	0	0	0	0	0		0	0	0	1	1	0	0	0
2	0 0 0 1 0	0 0	0	1	0	0	0	0	0		0	0	0	1	1	0	0	0
3	0 0 0 1 1	0 0	0	1	0	0	0	0	0		0	0	0	1	1	0	0	0
4	0 0 1 0 0	0 0	0	1	0	0	0	0	0		0	0	0	1	1	0	0	0
5	0 0 1 0 1	0 0	0	1	0	0	0	0	0		0	0	0	1	1	0	0	0
6	0 0 1 1 0	0 0	0	1	0	0	0	0	0		0	0	0	1	1	0	0	0
7	0 0 1 1 1	0 0	0	1	0	0	0	0	0		0	0	0	1	1	0	0	0
10	0 1 0 0 0	0 1	0	1	0	0	0	0	0		1	0	0	1	0	0	0	1
11	0 1 0 0 1	0 1	0	0	0	0	0	1	0		1	0	0	1	1	0	0	0
12	0 1 0 1 0	0 1	0	1	0	0	0	0	0		1	0	0	1	0	0	0	1
13	0 1 0 1 1	0 1	0	1	0	0	0	0	0		1	0	0	1	1	0	0	0
14	0 1 1 0 0	0 1	0	1	0	0	0	0	0		1	0	0	1	0	1	0	1
15	0 1 1 0 1	0 1	0	0	0	0	0	1	0		1	0	0	1	1	0	0	0
16	0 1 1 1 0	0 1	0	1	0	0	0	0	0		1	0	0	1	0	1	0	1
17	0 1 1 1 1	0 1	0	1	0	0	0	0	0		1	0	0	1	1	0	0	0
20	1 0 0 0 0	1 0	0	1	0	0	0	0	0		0	1	0	1	1	0	0	1
21	1 0 0 0 1	1 0	0	0	0	0	0	0	0		0	0	0	1	1	0	0	0
22	1 0 0 1 0	1 0	0	1	0	0	0	0	0		0	0	0	1	0	0	1	1
23	1 0 0 1 1	1 0	0	1	0	0	0	0	1		0	0	0	1	1	1	0	1
24	1 0 1 0 0	1 0	0	1	0	0	0	0	0		0	1	0	1	1	0	0	1
25	1 0 1 0 1	1 0	0	0	0	1	0	0	0		0	0	0	1	1	0	0	0
26	1 0 1 1 0	1 0	0	1	0	0	0	0	0		0	0	0	1	0	1	1	1
27	1 0 1 1 1	1 1	0	1	0	0	1	1	0		1	0	0	1	1	1	1	1
30	1 1 0 0 0	1 1	0	1	0	0	0	0	0		1	0	0	1	0	0	0	1
31	1 1 0 0 1	1 1	0	0	0	0	1	1	1		1	0	0	1	1	0	0	0
32	1 1 0 1 0	1 1	0	1	0	0	0	0	0		1	0	0	1	0	0	0	1
33	1 1 0 1 1	1 1	0	1	0	0	0	0	0		1	0	0	1	1	0	0	0
34	1 1 1 0 0	1 1	0	1	0	0	0	0	0		1	0	0	1	0	1	0	1
35	1 1 1 0 1	1 1	0	0	0	0	0	1	0		1	0	0	1	1	0	0	0
36	1 1 1 1 0	1 1	0	1	0	0	0	0	0		1	0	0	1	0	1	0	1
37	1 1 1 1 1	1 1	0	0	1	0	0	0	0		1	0	0	1	1	0	0	0

Table 3.1

INPUTCTLPROM and INPUTCTLREG

Address - Data Table

CTLREG. Thus it is low for addresses $(ØØ - 17)_8$ and high for addresses $(20 - 37)_8$.

Using table 3.1 the level of the control signals can be determined for all states of the Input Controller. Six of the key control signals and their functions are:

*SETDATOT (*SETDATAOUT) - The signal that informs the Line Generator that an input has been received from the FIFO and it should perform an operation on that data. It is used to set the DATAOUT flip flop which informs the Arithmetic Controller that the Input Controller is ready to pass it an input address.

FIFOINPUT - The signal that tells the FIFO that the L.G. is in state ØØ awaiting an input from the FIFO.

SETCHDATAOUT - This signal indicates that the Input Controller is in state 2Ø or 24. That a CHARACTER input command has been received from the FIFO is indicated by this signal. It is used to set the CHDATAOUT flip flop, see figure 3.3 and sheet 4 of the 195218 logic, It is one of the signals used to start the clock in the Character Generator and will be described in detail in the Character Generator description.

ACTIVEATELIGHTPEN - The signal that indicates a Light Pen input. High only in state 37. Used with the signal LIGHTPENAKG to generate a *CHANGEDATACTL. See figure 3.3 and sheet 4 of the 195218 logic.

WAITLOADSTATUR - Indicates the last state of the STATUS command input sequence, state 25. Used to insure that data is not changed until the new status is loaded.

*FIFOAKG - Indicates last state of an Input Control sequence. Used to K the FIFOREADY flip flop and make *LGSTR low, the acknowledge sent back to the FIFO. See figure and sheet 4 of the 195218 logic. As explained above it also disables the Input Controller multiplexer when low.

There are several other functions implemented on the 195218 card. These are described below in Section 4.8.

3.3 The Arithmetic Control State Machine (195216)

The following figures, drawings and tables as referenced in this description of the Arithmetic Controller.

a.	Figure 3.5	-	Arithmetic Control State Machine Block Diagram
b.	195216-600	-	Arithmetic Control Logic
c.	195216-950 Sheets 1 and 2	-	Arithmetic Control - Arith-Shifter Control Logic (State Diagram)
d.	Figure 3.6	-	Detailed State Diagram Arithmetic Control
e.	Table 3.2	-	ARITHCTLROM and ARITHCTLREG Address - Data Table

The Arithmetic Controller is a PROM implemented finite state machine that produces the control signals for the 195217 LINE GENERATOR X-Y ADDER SHIFTER.

A simplified block diagram of the Arithmetic Controller is shown in figure 3.5. The A input to the 2 to 1 multiplier comes off the AOPN bus. As explained above this bus has either the four OPN outputs of the INPUTCTLREG or the Character Generator DO IT REGISTER on it. (See figure 3.1 above). The A inputs are selected for the upper four bits of the ARITHCTLROM address when ARITHINPUT is high. The least address bit will be a Ø as * ARITHINPUT is low.

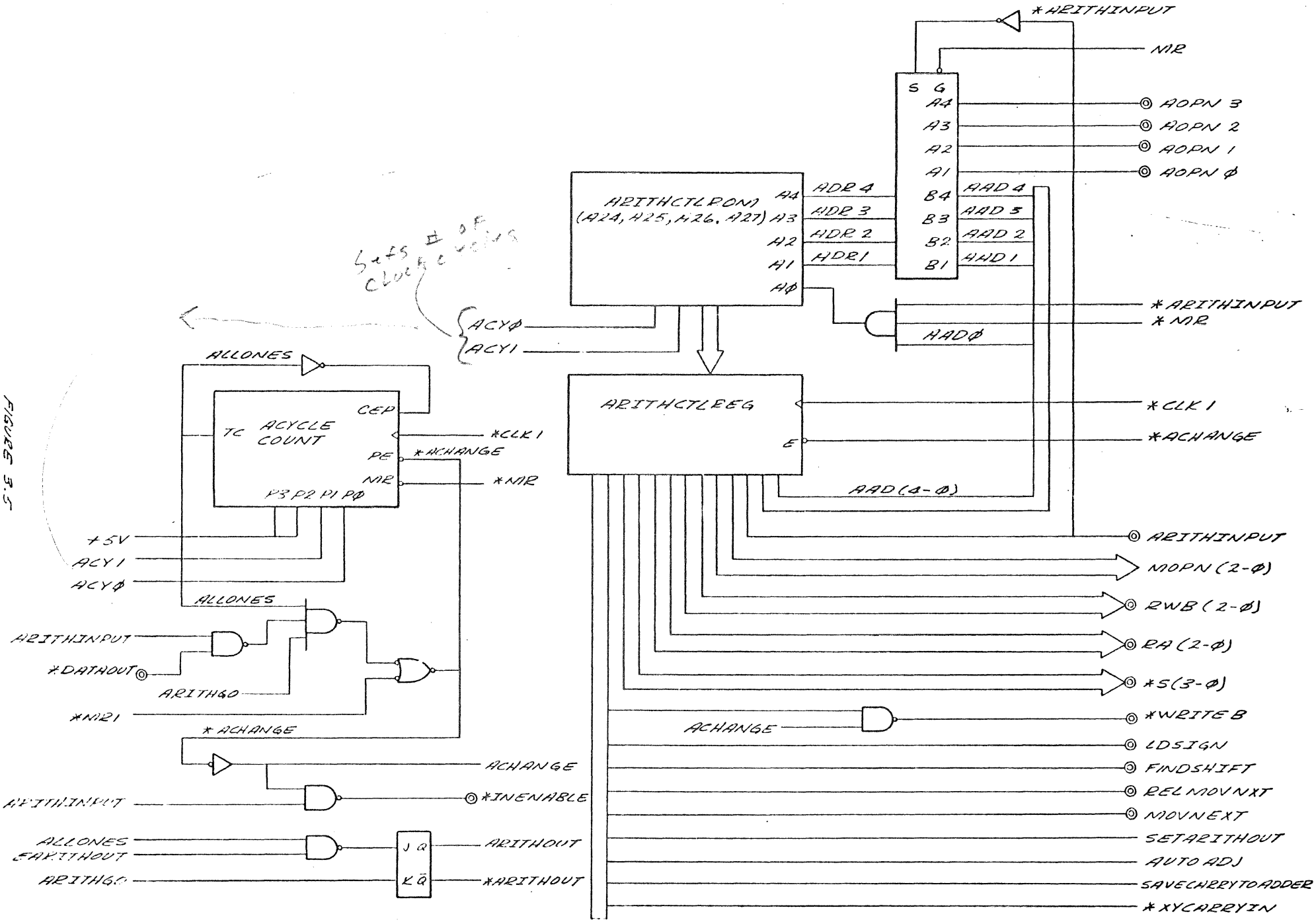


FIGURE 3.5
ARITHMETIC CONTROL STATE MACHINE BLOCK DIAGRAM
REF 193216 SHEETS 1 & 2

ARITHINPUT is an output of the ARITHCTLREG hence a ARITHCTLROM stored signal. The state diagram for the Arithmetic Control is shown on sheets 1 and 2 of the 195216-950 drawing and table 3.2 shows the contents of ARITHCTLROM and ARITHCTLREG. From the table it can be seen that ARITHINPUT is high, multiplexer selecting A inputs, for state ØØ and for all those states that are the last states of a sequence, Ø3, Ø5, Ø7, etc. Thus from state ØØ and the last state of a sequence the Arithmetic Controller will dispatch to the state specified by the input address received from the AOPN bus.

The outputs of the ARITHCTLROM are clocked into the ARITHCTLREG and into the ACYCLE COUNT, counter, when * ACHANGE is low. The ACYCLE COUNT counter is thus loaded with the binary number:

1 1 ACY1, ACYØ

ACYCLE COUNT is counted by the clock and * ACHANGE can not go low again until the counter overflows. ie. Its carry out, TC, which is the signal ALLONES, goes high. Thus ACYCLE COUNT determines the number of clock periods that a particular state exists and is the number given in the lower right corner of the state blocks on the state diagram. For example state Ø4. From the ARITHCTLROM table note that for state Ø4 ACY1=1 and ACYØ=Ø. Thus ACYCLE COUNT is loaded with the binary number 111Ø when ARITHCTLROM output is clocked into ARITHCTLREG. It requires two additional

ARITHMETIC CONTROL STATES (OCTAL)	A4 A3 A2 A1 A0	PROM A24 B7 B6 B5 B4 B3 B2 B1 B0: ACY1 ACY0 MOVNEXT RELMOVNXT MOPN2 MOPN1 MOPN0	PROM A25 B7 B6 B5 B4 B3 B2 B1 B0: *S3 *S2 *S1 *S0 *XYCARRYIN SAVECARRYTOADDER RA2 RA1	PROM A26 B7 B6 B5 B4 B3 B2 B1 B0: PA0 PWR2 PWR1 RWB0 72.13 (*WRITEB) AUTOADJ FINDSHIFT LDSIGN	PROM A27 B7 B6 B5 B4 B3 B2 B1 B0: SEARITHOUT ARITHINPUT AAD4 AAD3 AAD2 AAD1 AAD0
0	0 0 0 0 0	1 1 0 0 0 0 0 0	1 1 0 0 0 0 0 0	0 1 1 0 1 0 0 0	0 0 1 0 0 0 0 0
1	0 0 0 0 1	1 1 0 0 0 1 0 0	1 1 1 1 1 0 0 0	1 0 0 0 0 0 0 0	0 1 0 0 0 1 1 1
2	0 0 0 1 0	1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 1 0 0 0 0 1 1
3	0 0 0 1 1	1 1 0 0 0 0 0 1	1 1 1 1 1 0 0 0	1 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0
4	0 0 1 0 0	1 0 0 0 0 0 0 0	1 0 0 1 0 0 0 0	0 0 0 1 1 0 1 1	0 0 0 1 1 1 0 0
5	0 0 1 0 1	1 1 0 0 0 1 0 1	1 1 1 1 1 0 0 0	0 0 0 1 1 0 0 0	0 0 1 0 0 0 0 0
6	0 0 1 1 0	1 0 0 0 0 0 0 0	1 1 1 1 1 0 0 0	0 0 0 0 0 0 1 1	0 0 0 0 0 0 0 1
7	0 0 1 1 1	1 1 0 0 0 1 0 0	1 1 1 1 1 0 0 0	1 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0
8 0	0 1 0 0 0	1 0 0 0 0 0 0 0	0 1 1 0 1 0 0 0	1 0 0 0 1 0 0 0	0 0 0 0 1 0 0 1
8 1	0 1 0 0 1	1 0 0 0 0 0 0 0	1 0 0 1 0 0 0 0	0 0 0 1 1 0 1 1	0 1 0 1 0 1 0 1
8 2	0 1 0 1 0	1 0 0 0 0 0 0 0	0 1 1 0 1 0 1 0	1 1 1 0 1 0 0 1	0 0 0 0 1 0 1 1
8 3	0 1 0 1 1	1 0 0 0 0 0 0 0	0 1 1 0 1 1 1 0	0 0 0 0 1 0 0 0	0 0 0 0 1 0 0 0
8 4	0 1 1 0 0	1 0 0 0 0 0 0 0	0 1 1 0 1 0 0 0	1 0 0 0 1 0 0 0	0 0 0 0 0 1 0 0
8 5	0 1 1 0 1	1 1 0 0 0 0 0 0	0 1 1 0 1 0 0 0	0 0 0 1 1 0 0 0	0 1 0 1 0 1 1 1
8 6	0 1 1 1 0	1 0 0 0 0 0 0 0	1 0 0 1 0 0 1 1	1 0 0 1 0 0 0 1	0 0 0 0 1 1 1 1
8 7	0 1 1 1 1	1 1 1 0 0 0 0 0	1 1 1 1 1 0 0 0	1 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0
9 0	1 0 0 0 0	1 1 0 0 0 0 0 0	1 1 1 1 1 0 0 0	1 1 1 1 1 0 0 0	0 0 0 1 0 0 0 1
9 1	1 0 0 0 1	1 1 0 0 0 0 0 0	1 1 1 1 1 0 0 1	0 0 0 0 1 0 0 0	0 0 0 1 1 1 1 0
9 2	1 0 0 1 0	1 1 0 0 0 0 0 0	1 1 1 1 1 0 0 0	1 1 1 1 1 0 0 0	0 0 0 1 0 0 1 1
9 3	1 0 0 1 1	1 1 0 0 0 0 0 0	1 1 1 1 1 0 0 1	1 0 0 0 1 0 0 0	0 0 0 1 1 1 1 0
9 4	1 0 1 0 0	1 1 0 0 0 0 0 0	1 1 1 1 1 0 0 0	1 0 1 0 1 0 0 0	0 0 0 0 0 0 0 0
9 5	1 0 1 0 1	1 1 0 1 1 0 1 1	1 1 1 1 1 0 0 0	0 0 0 1 1 0 0 0	0 0 1 0 0 0 0 0
9 6	1 0 1 1 0	1 1 0 0 0 0 0 0	1 1 1 1 1 0 0 0	1 0 1 1 1 0 0 0	0 0 0 0 0 0 0 0
9 7	1 0 1 1 1	1 1 0 0 0 1 0 1	1 1 1 1 1 0 0 0	1 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0
10 0	1 1 0 0 0	1 1 0 0 0 0 0 0	1 1 1 1 1 0 0 0	1 0 0 0 0 0 0 0	0 1 0 1 1 1 0 1
10 1	1 1 0 0 1	1 1 0 0 0 0 0 0	1 1 0 0 0 0 0 0	0 1 1 0 1 0 0 0	0 1 0 1 1 1 1 1
10 2	1 1 0 1 0	1 1 0 0 0 0 0 0	1 1 1 1 1 0 1 1	1 0 0 0 1 0 0 0	0 0 0 1 1 0 1 0
10 3	1 1 0 1 1	1 0 0 0 0 0 0 0	1 0 0 1 0 0 1 1	1 0 0 1 1 0 1 1	0 1 0 0 0 0 0 1
10 4	1 1 1 0 0	1 1 0 0 0 1 0 1	1 1 1 1 1 0 0 0	0 0 0 1 0 0 0 0	0 1 0 0 0 1 0 1
10 5	1 1 1 0 1	1 1 0 0 0 0 1 0	1 1 1 1 1 0 0 0	1 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0
10 6	1 1 1 1 0	1 0 0 0 0 0 0 0	1 0 0 1 0 0 0 0	0 0 0 1 1 0 1 1	0 0 0 1 1 0 0 1
10 7	1 1 1 1 1	1 1 0 1 0 0 1 1	1 1 1 1 1 0 0 0	0 0 0 1 1 0 0 0	0 0 1 0 0 0 0 0

Table 3.2
ARITHCTLROM and ARITHCTLREG Address - Data Table

clock pulses to generate a carry out of ACYCLE COUNT and permit the next * ACHANGE to be generated.

When ARITHINPUT is low the multiplexer selects the B inputs, AAD4 thru AADØ from the ARITHCTLREG as the ARITHCTLROM address. The sequencing through a set of states can be traced by examining the ARITHCTLROM, ARITHCTLREG table. As an example let us trace the R MOV (RELATIVE MOVE) sequence of states; 1Ø, 11, 25. Assume that the machine is presently in state 1Ø. From the table it can be see that ACY1, ACYØ = 1Ø so that the machine will be in state 1Ø for two clock periods. ARITHINPUT = Ø thus the address presented to ARITHCTLROM is the B input to the multiplexer. This address is given by AAD4 thru AADØ of the table which for state 1Ø is $(11)_8$. Thus the machine will next dispatch to state 11. For state 11 ;

ACY1, ACYØ = 1, Ø	-	A two clock period state time
ARITHINPUT = Ø	-	ARITHCTLROM address is the B input to the multiplexer
AAD4-AADØ = 25	-	The next state is 25
For state 25 ;		
ACY1, ACYØ = 1,1	-	A one clock period state time
ARITHINPUT = 1	-	ARITHCTLROM address is the A input to the multiplexer. Thus the next state is determined by the address present on the AOPN bus inputs.

The individual state blocks of the 195216-950 state diagram define the arithmetic operations that are performed by the 195217 Line Generator X-Y Adder Shifter card. Data transfer operations and the enabling of data onto busses are denoted by an arrow. For example the statement:

$$A \leftarrow XYIN$$

states that the quantity XYIN is enabled onto the A buss. The definition of the data busses and the data register are given on the 195216-950 State Diagram and sheet 1 of the data flow block diagram Figure 2.3 of Section 2.3 above.

The busses A, B and F refer to the two inputs and the output of the arithmetic logic elements, ALU's, see figure 2.3. The four functions performed by the ALU and the corresponding code required for the four control bits *S3 - *SØ and the carry input to perform that function are summarized in the ARITH ADDER CTL table shown on sheet 1 of the 195216-950 state diagram. The ALU performs four functions:

F ← A	A input buss passed to output F.
F ← A + B	F output is the arithmetic sum of A plus B
F ← A - B	F output is the arithmetic difference of A minus B
F ← Ø	F output is zero.

The data operated on by the X-Y Shifter Adder is stored in the X and the Y Register Files which

were loaded from the FIFO's via the XCBUS and YCBUS. The MULTIPORT REGISTER ASSIGNMENT table on sheet 1 of the 195216-950 state diagram shows the data stored in this 8 word memory file and its address. The data used in the line generator draw and move operations are XYIN, address Ø and XY, address 1. The remainder of the data are used when characters are drawn and are stored at locations 2 thru 8. The line drawing and move operations will be described in this section, the character related operations are described in the sections on the Character Generator below.

The following state sequences, are concerned with line drawing and moves:

DRAW (ABSOLUTE DRAW AUTO FIX)	- Ø4, 34, Ø5
R DRAW (RELATIVE DRAW AUTO FIX)	- 14, Ø4, 34, Ø5
MOV (ABSOLUTE MOVE)	- 36, 31, 37
R MOVE (RELATIVE MOVE)	- 10, 11, 25

Figure 3.6 is a more detailed state diagram which contains a brief descriptive paragraph next to each state box.

An understanding of the operation of the Arithmetic Controller and the associated X and Y Shifter Address is best acquired by tracing through the operations performed in a sequence of states. Consider as an example the absolute DRAW, states Ø4, 34, Ø5.

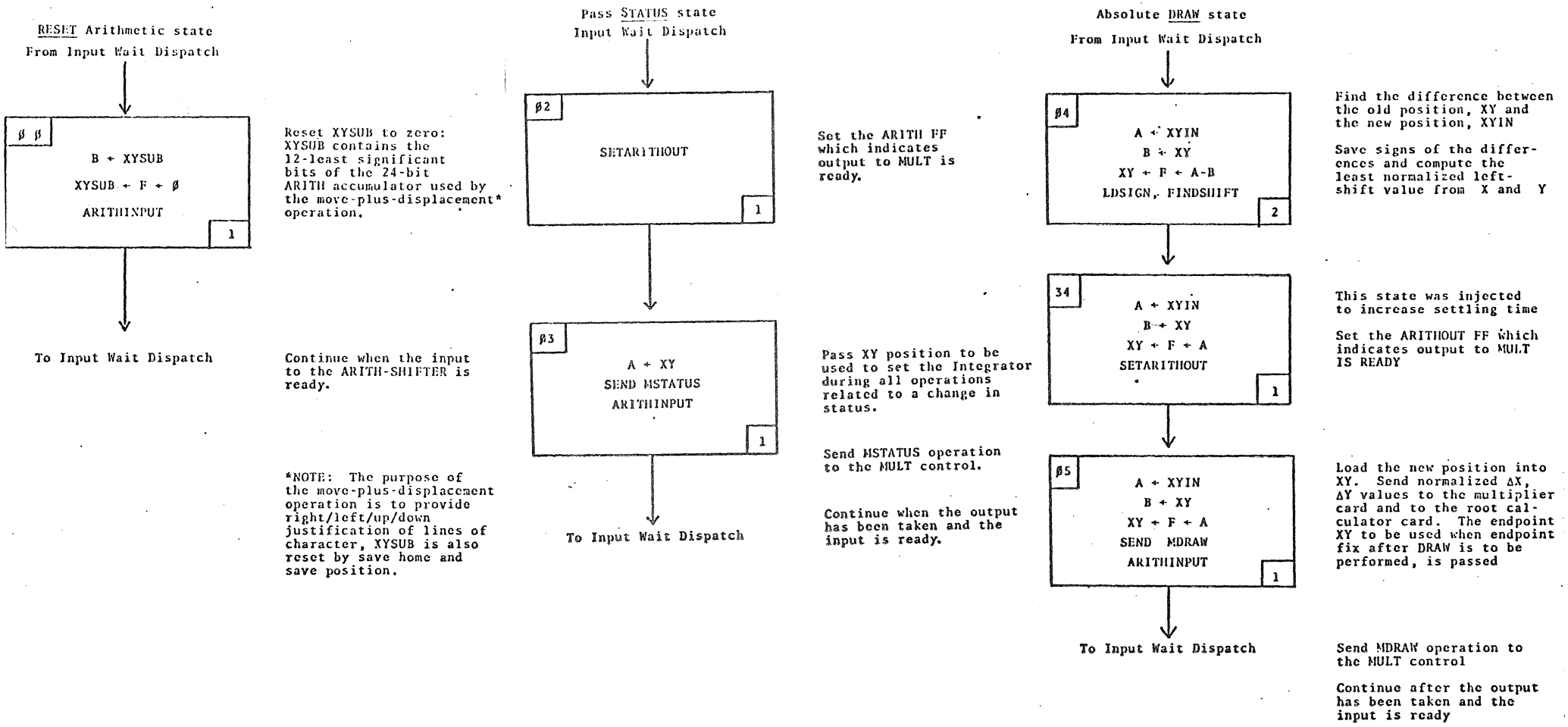

RESET Arithmetic state
From Input Wait Dispatch
ø ø
B ← XYSUB
XYSUB ← F ← ø
ARITHINPUT
1
To Input Wait Dispatch
Reset XYSUB to zero: XYSUB contains the 12-least significant bits of the 24-bit ARITH accumulator used by the move-plus-displacement* operation.
Continue when the input to the ARITH-SHIFTER is ready.
*NOTE: The purpose of the move-plus-displacement operation is to provide right/left/up/down justification of lines of character, XYSUB is also reset by save home and save position.
Pass STATUS state
Input Wait Dispatch
ø2
SETARITHOUT
1
ø3
A ← XY
SEND MSTATUS
ARITHINPUT
1
To Input Wait Dispatch
Set the ARITH FF which indicates output to MULT is ready.
Pass XY position to be used to set the Integrator during all operations related to a change in status.
Send MSTATUS operation to the MULT control.
Continue when the output has been taken and the input is ready.
Absolute DRAW state
From Input Wait Dispatch
ø4
A ← XYIN
B ← XY
XY ← F ← A-B
LDSIGN, FINDSHIFT
2
34
A ← XYIN
B ← XY
XY ← F ← A
SETARITHOUT
1
ø5
A ← XYIN
B ← XY
XY ← F ← A
SEND MDRAW
ARITHINPUT
1
To Input Wait Dispatch
Find the difference between the old position, XY and the new position, XYIN
Save signs of the differences and compute the least normalized left-shift value from X and Y
This state was injected to increase settling time
Set the ARITHOUT FF which indicates output to MULT IS READY
Load the new position into XY. Send normalized ΔX, ΔY values to the multiplier card and to the root calculator card. The endpoint XY to be used when endpoint fix after DRAW is to be performed, is passed
Send MDRAW operation to the MULT control
Continue after the output has been taken and the input is ready

Figure 3.6

DETAILED STATE DIAGRAM ARITHMETIC CONTROLLER

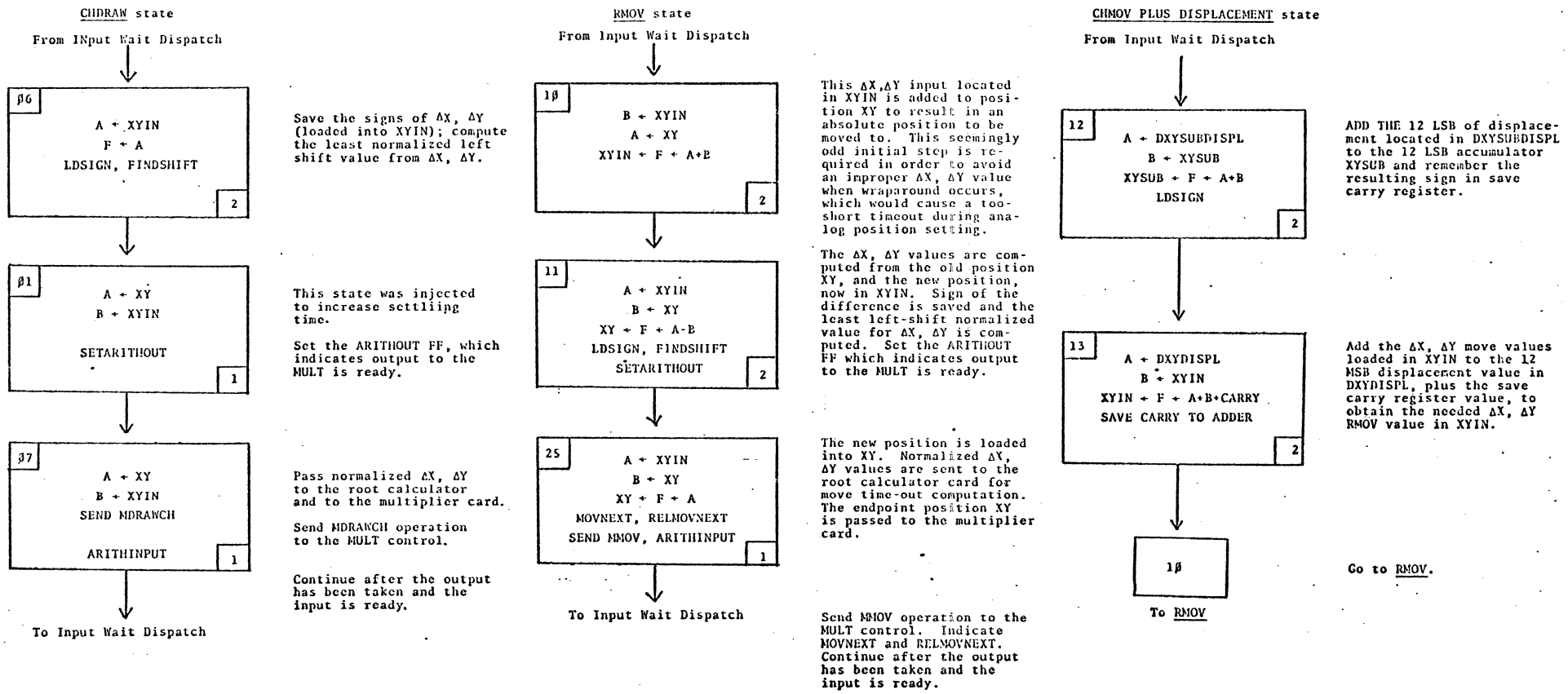

CHDRAW state
From INput Wait Dispatch
ØG
A ← XYIN
F ← A
LDSIGN, FINDSHIFT
2
Save the signs of ΔX, ΔY (loaded into XYIN); compute the least normalized left shift value from ΔX, ΔY.
Ø1
A ← XY
B ← XYIN
SETARITHOUT
1
This state was injected to increase settliing time.
Set the ARITHOUT FF, which indicates output to the MULT is ready.
37
A ← XY
B ← XYIN
SEND MDRAWCH
ARITHINPUT
1
Pass normalized ΔX, ΔY to the root calculator and to the multiplier card.
Send MDRAWCH operation to the MULT control.
Continue after the output has been taken and the input is ready.
To Input Wait Dispatch
RMOV state
From Input Wait Dispatch
1Ø
B ← XYIN
A ← XY
XYIN ← F ← A+B
2
This ΔX,ΔY input located in XYIN is added to position XY to result in an absolute position to be moved to. This seemingly odd initial step is required in order to avoid an improper ΔX, ΔY value when wraparound occurs, which would cause a too-short timeout during analog position setting.
11
A ← XYIN
B ← XY
XY ← F ← A-B
LDSIGN, FINDSHIFT
SETARITHOUT
2
The ΔX, ΔY values are computed from the old position XY, and the new position, now in XYIN. Sign of the difference is saved and the least left-shift normalized value for ΔX, ΔY is computed. Set the ARITHOUT FF which indicates output to the MULT is ready.
25
A ← XYIN
B ← XY
XY ← F ← A
MOVNEXT, RELMOVNEXT
SEND MMOV, ARITHINPUT
1
The new position is loaded into XY. Normalized ΔX, ΔY values are sent to the root calculator card for move time-out computation. The endpoint position XY is passed to the multiplier card.
Send MMOV operation to the MULT control. Indicate MOVNEXT and RELMOVNEXT. Continue after the output has been taken and the input is ready.
To Input Wait Dispatch
CHMOV PLUS DISPLACEMENT state
From Input Wait Dispatch
12
A ← DXYSUBDISPL
B ← XYSUB
XYSUB ← F ← A+B
LDSIGN
2
ADD THE 12 LSB of displacement located in DXYSUBDISPL to the 12 LSB accumulator XYSUB and remember the resulting sign in save carry register.
13
A ← DXYDISPL
B ← XYIN
XYIN ← F ← A+B+CARRY
SAVE CARRY TO ADDER
2
Add the ΔX, ΔY move values loaded in XYIN to the 12 MSB displacement value in DXYDISPL, plus the save carry register value, to obtain the needed ΔX, ΔY RMOV value in XYIN.
1Ø
Go to RMOV.
To RMOV

Figure 3.6

DETAILED STATE DIAGRAM ARITHMETIC CONTROLLER

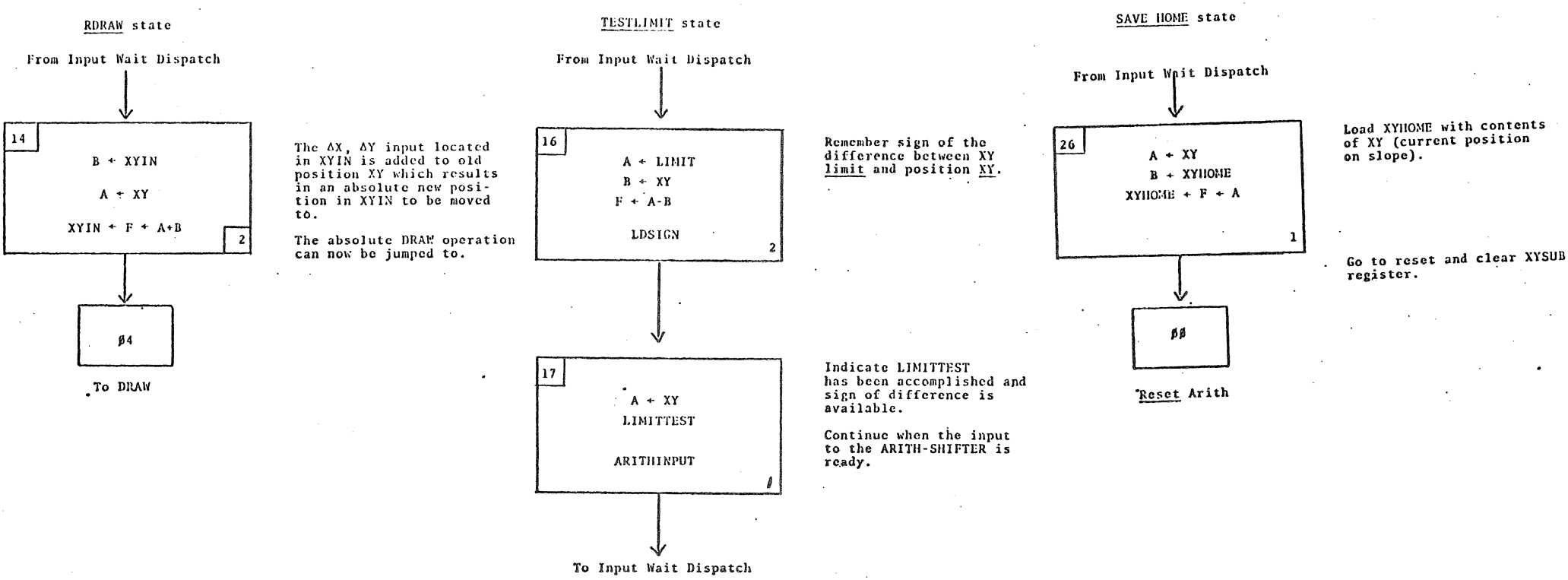

RDRAW state
From Input Wait Dispatch
14
B ← XYIN
A ← XY
XYIN ← F ← A+B
2
ø4
To DRAW
The ΔX, ΔY input located in XYIN is added to old position XY which results in an absolute new position in XYIN to be moved to.
The absolute DRAW operation can now be jumped to.
TESTLIMIT state
From Input Wait Dispatch
16
A ← LIMIT
B ← XY
F ← A-B
LDSIGN
2
17
A ← XY
LIMITTEST
ARITHINPUT
1
To Input Wait Dispatch
Remember sign of the difference between XY limit and position XY.
Indicate LIMITTEST has been accomplished and sign of difference is available.
Continue when the input to the ARITH-SHIFTER is ready.
SAVE HOME state
From Input Wait Dispatch
26
A ← XY
B ← XYHOME
XYHOME ← F ← A
1
øø
Reset Arith
Load XYHOME with contents of XY (current position on slope).
Go to reset and clear XYSUB register.

Figure 3.6

DETAILED STATE DIAGRAM ARITHMETIC CONTROLLER

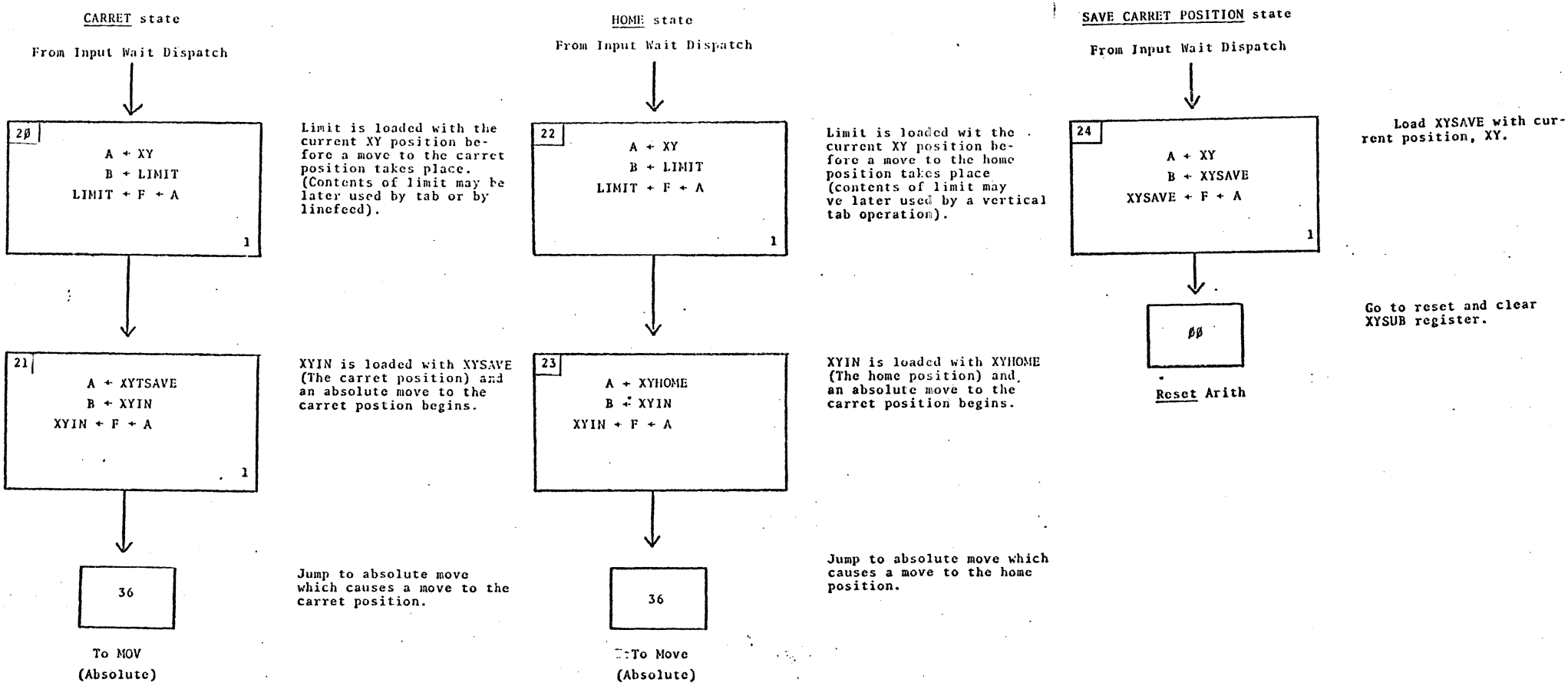

CARRET state
From Input Wait Dispatch
20
A ← XY
B ← LIMIT
LIMIT ← F ← A
1
Limit is loaded with the current XY position before a move to the carret position takes place. (Contents of limit may be later used by tab or by linefeed).
21
A ← XYTSAVE
B ← XYIN
XYIN ← F ← A
1
XYIN is loaded with XYSAVE (The carret position) and an absolute move to the carret postion begins.
36
Jump to absolute move which causes a move to the carret position.
To MOV
(Absolute)
HOME state
From Input Wait Dispatch
22
A ← XY
B ← LIMIT
LIMIT ← F ← A
1
Limit is loaded wit the current XY position before a move to the home position takes place (contents of limit may ve later used by a vertical tab operation).
23
A ← XYHOME
B ← XYIN
XYIN ← F ← A
XYIN is loaded with XYHOME (The home position) and an absolute move to the carret position begins.
36
Jump to absolute move which causes a move to the home position.
To Move
(Absolute)
SAVE CARRET POSITION state
From Input Wait Dispatch
24
A ← XY
B ← XYSAVE
XYSAVE ← F ← A
1
Load XYSAVE with current position, XY.
00
Go to reset and clear XYSUB register.
Reset Arith

Figure 3.6

DETAILED STATE DIAGRAM ARITHMETIC CONTROLLER

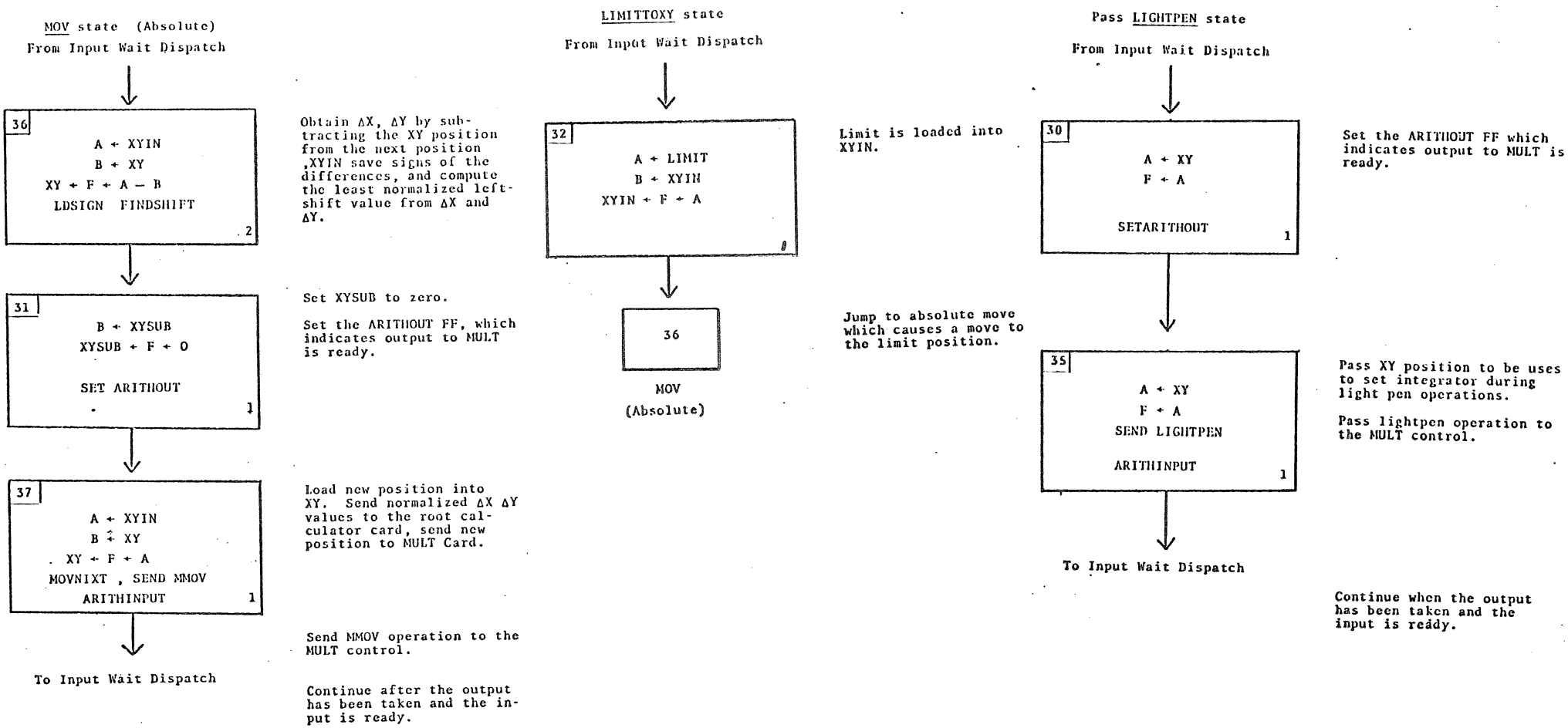

MOV state (Absolute)
From Input Wait Dispatch
36
A ← XYIN
B ← XY
XY ← F ← A − B
LDSIGN FINDSHIFT
2
Obtain ΔX, ΔY by subtracting the XY position from the next position ,XYIN save signs of the differences, and compute the least normalized left-shift value from ΔX and ΔY.
31
B ← XYSUB
XYSUB ← F ← 0
SET ARITHOUT
1
Set XYSUB to zero.
Set the ARITHOUT FF, which indicates output to MULT is ready.
37
A ← XYIN
B ← XY
XY ← F ← A
MOVNIXT , SEND MMOV
ARITHINPUT
1
Load new position into XY. Send normalized ΔX ΔY values to the root calculator card, send new position to MULT Card.
Send MMOV operation to the MULT control.
Continue after the output has been taken and the input is ready.
To Input Wait Dispatch
LIMITTOXY state
From Input Wait Dispatch
32
A ← LIMIT
B ← XYIN
XYIN ← F ← A
0
Limit is loaded into XYIN.
36
Jump to absolute move which causes a move to the limit position.
MOV
(Absolute)
Pass LIGHTPEN state
From Input Wait Dispatch
30
A ← XY
F ← A
SETARITHOUT
1
Set the ARITHOUT FF which indicates output to MULT is ready.
35
A ← XY
F ← A
SEND LIGHTPEN
ARITHINPUT
1
Pass XY position to be uses to set integrator during light pen operations.
Pass lightpen operation to the MULT control.
To Input Wait Dispatch
Continue when the output has been taken and the input is ready.

Figure 3.6

DETAILED STATE DIAGRAM ARITHMETIC CONTROLLER

STATE Ø4

The absolute coordinates, X and Y of the new point, the point to be drawn to, are read from the X and the Y register file locations XYIN. The coordinates of the old point are read from register file locations XY. XYIN is applied to the A input and XY to the B input of the ALU. This is shown in the state diagram by the statement A ← XYIN and B ← XY.

Referring to state Ø4 of the ARITHCTLROM and ARITHCTLREG Address - Date table we find that location Ø, XYIN is being read out to the A input of the ALU and location 1, XY, is being read out to the B input. This is shown by the register file read address for the A input to the ALU being RA2, RA1, RAØ=ØØØ and the register file read address for the B input RWB2, RWB1, RWBØ=ØØ1. The statement XY ← F ← A-B says that XYIN HAS XY subtracted from it to form ΔX and ΔY. The ALU function code for state Ø4 in the table is *S3, *S2 *S1 *SØ = 10Ø 1, the subtract function. Note that the signal *WRITEB of the table is a 1 which means that the F output of the ALU, which is A-B, is not written into the register file as the write enable, *WRITE B, is high. The other control signals that are active during state Ø4 are: (by active it is meant a * control signal is low and a non * is high)

LDSIGN - Enables the clock to SIGN flip flop that stores the signs of ΔX and ΔY. These signs are

needed by the normalizing logic. The SIGN flip flop is shown on sheet 3 of the 195217 logic, I.C. position 1Ø).

FINDSHIFT - Enables the ENCODE multiplexer that selects the ENCODE number generated by the normalization of the X and Y on the 195217. (See sheet 3, 195217 logic, I.C. 34 and 11).

SAVECARRYTOADDER - This signal is low in state Ø4. It selects *XCARRYOUT and *YCARRYOUT as the carry inputs *XCARRYIN and *YCARRYIN to the ALU's. This selector is shown on sheet 4 of the 195216 logic.

Note that ARITHINPUT is low and the AAD4 - AADØ outputs of ARITHCTLREG are selected as the address inputs to ARITHCTLROM. AAD4 - AADØ = 34 thus the next state is 34.

STATE 34

In state 34 the A input and B input to the ALU 's are the same as in state Ø4. RA2 - RAØ and RWB2 - RWBØ are the same. The ALU function code *S3 - *SØ is 1111 thus the F output of the ALU is the A input as is indicated by the statement F A. Note that this statement also says that XY F A. As A is XYIN the new point XYIN is read into the old point register file location XY. It will become the old point for the next DRAW or MOVE instruction executed. The fact that * WRITE B is low for state 34 means that the location RWB2 - RWBØ = 001, location of XY, is written into. The other active signal of note is SETARITHOUT which is high. This signal allows the ARITHOUT flip flop

to be set which indicates to the Multiplier Controller that the Arithmetic Controller has the MOPN(2-Ø) bits, the address for MULTCTLROM ready for it. The AAD4 - AADØ code ØØØ1Ø1, ARITHCTLROM address, puts the machine into state Ø5. Also note that the address bits, MOPN2-MOPNØ, passed on to the Multiplier Controller have the binary code 101. This code is set during state 34 but not "passed" to the Multiplier Controller. This code will also be set during the next state, state Ø5.

STATE Ø5

The ALU's perform the same function as they did in the previous state, state 34. Thus the ALU function bits *S3 - *SØ are the same. The register file read address bits RA2 - RAØ and RWB2 - RWBØ are also the same. However note that the write enable *WRITE B is now high and the ALU output F is thus not written into the register file as it was in state 34. The ARITHINPUT bit is a 1 which means that the ARITHCTLROM address will be selected from the AOPN bus. During states Ø4 and 34 above ARITHINPUT was low so that the selected address bits were the AAD4-AADØ outputs of ARITHCTLREG. The MOPN2-MOPNØ bits are set to the binary value 1Ø1. These bits are the top three bits of the address input to the Multiplier Controllers PROM whose operation will be described below.

The reader should trace the operations of R DRAW, MOV and RMOV through the Arithmetic Control states to insure that he understands the function of the Arithmetic Control.

The function of the control signals shown on the Arithmetic Control Logic block diagram are summarized below.

AAD4 - AADØ	-	Address bits fed back to address ARITHCTLROM when *ARITHINPUT is high.
ARITHINPUT	-	When high indicates that the Arithmetic Controller is ready to start a new sequence of states, it selects the AOPN bus inputs as the address for the ARITCLTROM and insures that the least address bit is zero.
MOPN2-MOPNØ	-	The address bits for the MULTCTLROM on the 195216, ARITH CONTROL card. These bits are applied to a 2/1 multiplexer and selected as the PROM address when *LOADM is low. See sheet 3.
RWB2-RWBØ	-	The write/read address for section 2 (or B) of the multiport register file of the 195217 ADDER SHIFTER. Section 2 allows reading or writing or simultaneous read and write of the same location.
RA2 - RAØ	-	The Read Address for Section 1 (or B) the multiport register file of the 195217 ADDER SHIFTER. Section 1 allows writing of data into any word while simultaneously reading another location.
*S3 - *SØ	-	The ALU control bits for the 195217 X-Y ADDER SHIFTER. Determine the function performed by the ALU. (See the ARITH ADDER CTL table on sheet 1 of the 195216-950 Arithmetic Control State diagram)
*WRITE B	-	Enables writing into section 2 (or B) of the register file on the XY ADDER SHIFTER cards, 195217's, sheet 1.
LDSIGN	-	When high enables SIGN flip flop of the 195217's, sheet 3, to be clocked.
FINDSHIFT	-	When high enables the ENCODE multiplexer on the 195217's, sheet 3. Enables the clock to the SHIFT latch on the ROOT CALCULATOR, 195215 sheet 2, I.C. 20.9.
RELMOUNXT	-	Modifies the Character Overflow ROM address, CHARACTER CONTROL card, 195223 sheet 4, IC 63.2, to cause a relative move during the next state when clipping is used.

MOVNEXT	-	Indicates that the next command will be a MOVE. Employed as address bit Ø on the SPEED CONTROL ROM of the OUTPUT CONTROL, 195213, sheet 4, IC 26.5.
SETARITOUT	-	'ANDED' with the carry output, AALONES, of the ACYCLECOUNT counter to set the ARITHOUT flip flop, sheet 2, 195216 logic.
AUTOADJ	-	Unused
SAVECARRYTOADDER	-	When high selects *XCARRYOUT and *YCARRYOUT, see IC 26 sheet 4, as the carry inputs, *XCARRYIN and *YCARRYIN, to the ALU's on the ADDER SHIFTER'S, 195217's, sheet 1.
*XYCARRYIN	-	Is selected as the *XCARRYIN and *YCARRYIN by the selector, IC 26, sheet 4 when SAVECARRYTOADDER is low.
ACY1, ACYØ	-	Preset lower two bits of the arithmetic cycle counter, ACYCLECOUNT, sheet 2 IC 52.3, 52.4.
*INENABLE	-	When low enables the 2/1 multiplexer that selects the signals *INENABLEX and *INENABLEY which are the write enables for section 1 of the register file on the 195217's. The 2/1 multiplexer is located on the INPUT CONTROL Card 195218, sheet 4. Also enables the flip flop shown on sheet 6 of the CHARACTER MULTIPLIER, 195221 car
*DATAOUT	-	Received from the DATAOUT flip flop on the 195218 Input Controller. Indicates that the Input Controller has address bits for the Arithmetic Controller. (See 195218 logic, sheet 4)
ARITHGO	-	Control signal that resets the ARITHOUT flip flop when the Multiplier Controller takes the MOPN address bits. Also must be high to generate *ACHANGE and let the Arithmetic Controller "GO" to its next state.

The Multiplier Controller is also implemented on the 195216 and is described in Section 3.4 which follows.

3.4 The Multiplier Control State Machine (195216)

The Multiplier Control State Machine is also implemented on the 195216 card. The following figure, drawings, and tables are necessary to understanding its operation.

a.	Figure 3.7	-	Multiplier Control State Machine Block Diagram
b.	195216-600	-	Arithmetic Control Logic Diagram
c.	195216-950	-	Sheet 3 - Multiplier Control State Diagram
d.	Figure 3.8	-	Detailed State Diagram Multiplier Controller
e.	Table 3.3	-	MULTCTLROM and MULTCTLREG Address Data Table

Figure 3.7 is a simplified block diagram of the Multiplier Control logic. Sheet 3 of the 195216-950 Arithmetic Control drawing is the state diagram for the Multiplier Controller and Table 3.3 shows the MULTCTLROM-MUTLCTLREG Address-Data. The Multiplier Controller is similar in operation to the Input Controller and the Arithmetic Controller. The reader should be able to trace the operation using the State Diagram and the Address Table if he has read the previous two sections which detailed this process for the Input and Arithmetic Controller.

The address for the MULTCTLROM is obtained from a 2 to 1 MULTIPLEXER. When *LOADM is low the MULTCTLROM address is;

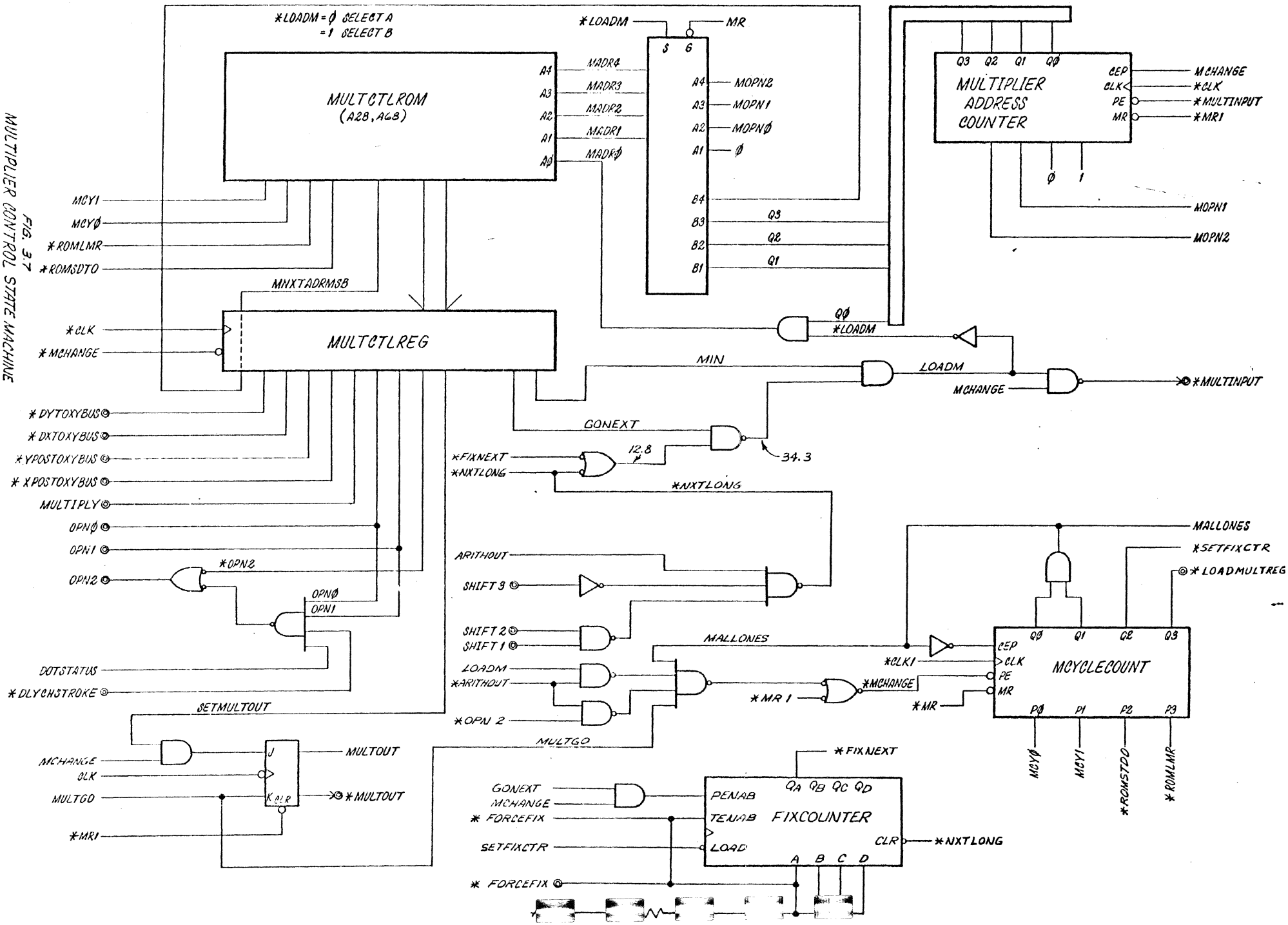


FIG. 3.7
MULTIPLIER CONTROL STATE MACHINE
BLOCK DIAGRAM
REF. 195816 SHEET 3

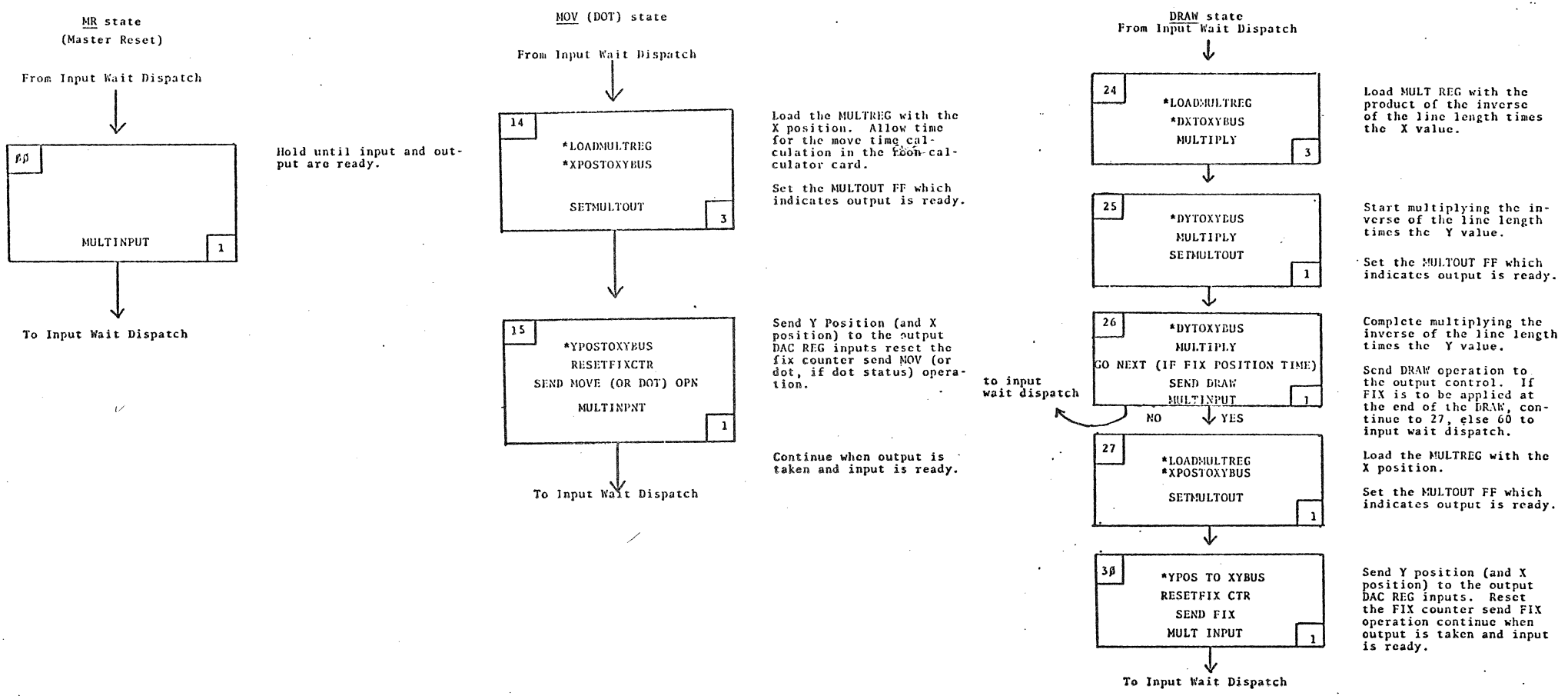

MR state
(Master Reset)
From Input Wait Dispatch
00
MULTINPUT
1
To Input Wait Dispatch
Hold until input and output are ready.
MOV (DOT) state
From Input Wait Dispatch
14
*LOADMULTREG
*XPOSTOXYBUS
SETMULTOUT
3
Load the MULTREG with the X position. Allow time for the move time calculation in the 100n calculator card.
Set the MULTOUT FF which indicates output is ready.
15
*YPOSTOXYBUS
RESETFIXCTR
SEND MOVE (OR DOT) OPN
MULTINPNT
1
Send Y Position (and X position) to the output DAC REG inputs reset the fix counter send MOV (or dot, if dot status) operation.
Continue when output is taken and input is ready.
To Input Wait Dispatch
DRAW state
From Input Wait Dispatch
24
*LOADMULTREG
*DXTOXYBUS
MULTIPLY
3
Load MULT REG with the product of the inverse of the line length times the X value.
25
*DYTOXYBUS
MULTIPLY
SETMULTOUT
1
Start multiplying the inverse of the line length times the Y value.
Set the MULTOUT FF which indicates output is ready.
26
*DYTOXYBUS
MULTIPLY
GO NEXT (IF FIX POSITION TIME)
SEND DRAW
MULTINPUT
1
Complete multiplying the inverse of the line length times the Y value.
Send DRAW operation to the output control. If FIX is to be applied at the end of the DRAW, continue to 27, else 60 to input wait dispatch.
to input wait dispatch
NO
YES
27
*LOADMULTREG
*XPOSTOXYBUS
SETMULTOUT
1
Load the MULTREG with the X position.
Set the MULTOUT FF which indicates output is ready.
30
*YPOS TO XYBUS
RESETFIX CTR
SEND FIX
MULT INPUT
1
Send Y position (and X position) to the output DAC REG inputs. Reset the FIX counter send FIX operation continue when output is taken and input is ready.
To Input Wait Dispatch

Figure 3.8

DETAILED STATE DIAGRAM MULTIPLIER CONTROLLER

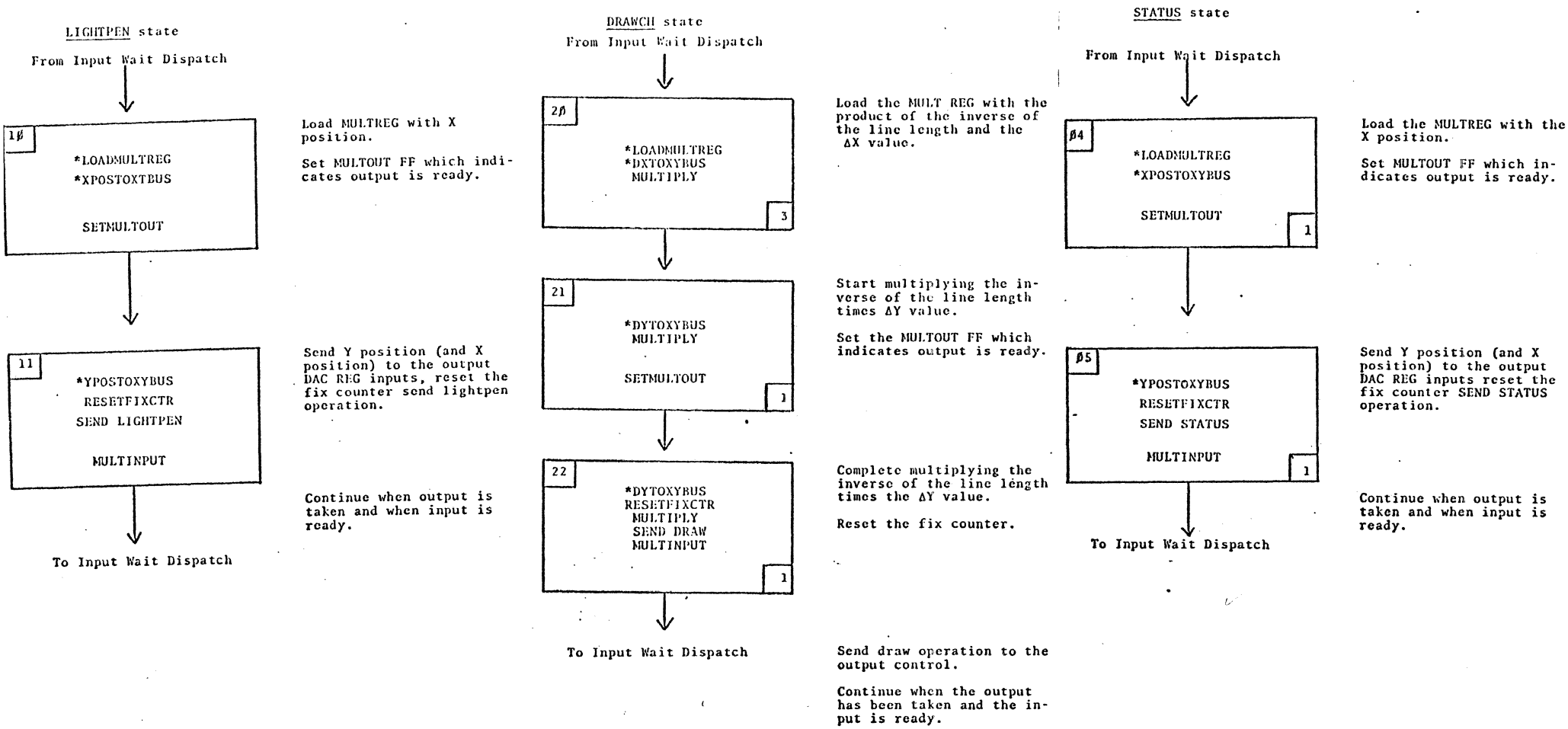


Figure 3.8

DETAILED STATE DIAGRAM MULTIPLIER CONTROLLER

	MULTIPLIER CONTROL STATES (OCTAL)	MULTCTLROM ADDRESS A4 A3 A2 A1 A0	PROM A28 B7 B6 B5 B4 B3 B2 B1 B0 (MCYC1) P1	(MCYC0) P0	SETMULTOUT	*DYTOXYBUS	*DXTOXYBUS	*YPOSTOXYBUS	*XPOSTOXYBUS	MULTIPLY	PROM A68 B7 B6 B5 B4 B3 B2 B1 B0 *LOADMULTREG	*SETFIXCTR	GONEXT	*OPN2	OPN1	OPN0	MIN	(MNXTADRMSB)
MR	0	0	1 1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	0
	1	1	0 0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0
	2	2	0 0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0
	3	3	0 0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0
STATUS	4	4	1 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	5	5	1 1	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0
	6	6	1 1	1	0	0	1	1	1	0	1	1	0	0	0	0	0	0
	7	7	1 1	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0
LIGHTPEN	10	8	1 0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
	11	9	1 0	0	1	1	1	0	1	0	1	1	0	0	1	0	1	0
	12	10	1 0	1	0	0	0	1	1	0	1	1	0	0	0	0	0	0
	13	11	1 0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0
MOV/DOT	14	12	1 1	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0
	15	13	1 1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0
	16	14	1 1	1	0	0	0	1	1	0	1	1	0	0	0	0	0	0
	17	15	1 1	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0
DRAWCH	20	16	1 0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	1
	21	17	1 0	0	0	1	1	1	0	1	1	1	1	0	0	0	0	1
	22	18	1 0	0	1	0	1	1	0	1	1	1	1	0	0	0	1	0
	23	19	1 0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0
DRAW	24	20	1 0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	1
	25	21	1 0	1	0	1	1	1	0	1	1	1	1	1	0	0	0	1
	26	22	1 0	1	1	0	0	1	0	1	1	1	1	1	0	0	1	1
	27	23	1 0	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1
	30	24	1 1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1
	31	25	1 1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0
	32	26	1 1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0
	33	27	1 1	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0
	34	28	1 1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
	35	29	1 1	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0
	36	30	1 1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
	37	31	1 1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0

Table 3.3
MULTCTLROM and MULTCTLREG
Address - Data Table

MOPN2, MONP1, MOPNØ, Ø, Ø

The MOPN inputs are the address bits passed forward by the Arithmetic Controllers ARITCTLREG. When *LOADM is high the address is selected from the MULTIPLIER ADDRESS COUNTER with the most significant bit coming from the MULTCTLREG which is the MULTCTLROM stored bit MNXTADRMSB. The MULTIPLIER ADDRESS COUNTER is loaded with the initial count

MOPN2, MOPN1, Ø, 1

when *MULTINPUT is low. *MULTINPUT is low if MCHANGE and LOADM are both high. MCHANGE is the state change logic output; its compliment *MCHANGE enables the loading of MULTCTL-REG and MCYCLECOUNT. Because of the initial count loaded into the MULTIPLIER ADDRESS COUNTER when the first state of a sequence is dispatched into the succeeding states in that sequence have sequential binary state numbers or MULTCTLROM addresses.

The MCYCLECOUNT counter determines the number of clock periods for which the controller remains in a particular state. Only its lower two bits which are loaded with MULTCTLROM outputs MCY1 and MCYØ, are used to time state duration. The state diagram shows that three clock periods, the number in the lower right of the state "blocks", is the maximum state duration for any of the states. MALLONES the logical AND of the

least two counter bits indicates that the machine has been in a particular state the required number of clock periods. The upper two bits of the MCYCLECOUNT counter, Q2 and Q3 are used as flip flops and store the MULTCTLROM outputs *ROMSDTO and *ROMLMR which become *SETFIXCTR and *LOADMULTREG at the output of MCYCLECOUNT.

The logical equation that defines when the state change signal *MCHANGE goes low is:

MALLONES . MULTGO . (*LOADM+ARITHOUT) . (OPN2+ARITHOUT)

MALLONES is the output of MCYCLECOUNT. MULTGO is a control signal that indicates that the states of the Line Generator controllers are in a condition that permits a Multiplier Control state change to occur, ARITHOUT is the signal that says the Arithmetic Controller has the address bits that it passes to the Multiplier Controller, MOPN (2-Ø), on their output lines. *LOADM will be high when the Multiplier Controller is not in the last state of a sequence of states and OPN2 is one of the address bits passed onto the Output Controller and is stored in MULTCTLROM.

The logic for generating the LOADM signal which selects the address for MULTCTLROM is the implementation of the logical equation:

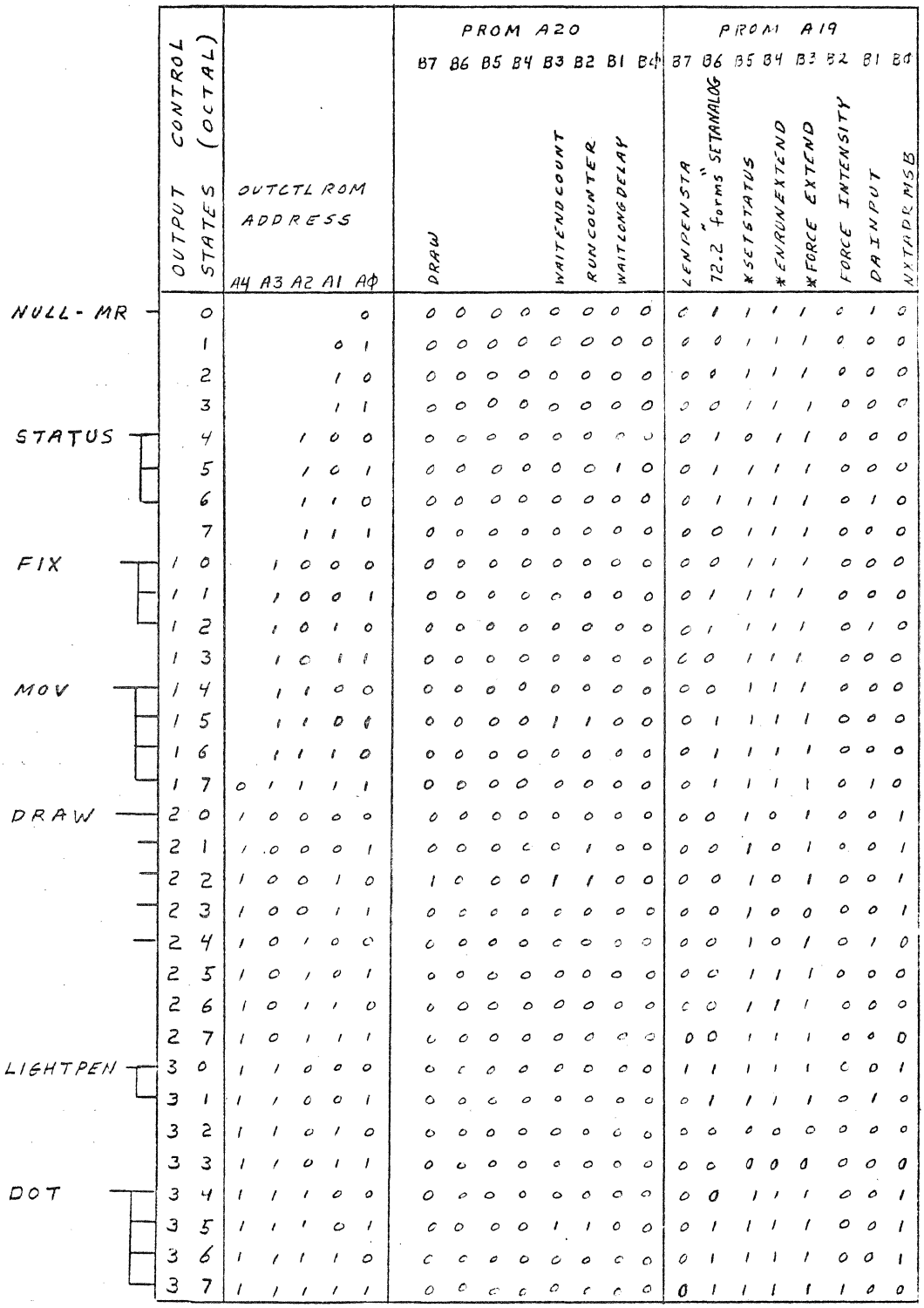
LOADM = MIN . (*GONEXT + *FIXNEXT . *NXTLONG)

when LOADM is high the selected MULTCTLROM address bits are the MOPN (2-Ø) bits passed over by the Arithmetic Controller.

MIN is a MULTCTLROM stored bit which is high for the last state of each sequence of states. See Table 3.4. Note that MIN is also high in state 26 from which, as indicated in the state diagram, a dispatch back to the first state of a sequence is possible.

GONEXT is high only in state 26. Therefore the branch taken out of state 26, either back to a new sequence of states or onto states 27 and 30, is determined by FIXNEXT and NXTLONG. If in state 26, where GONEXT is high, FIXNEXT or NXTLONG are high LOADM will be low and *LOADM high. Thus the address presented to MULTCTLROM is from the MULTIPLIER ADDRESS COUNTER. Because LOADM is low, a high *MULTINPUT is applied to the MULTIPLIER ADDRESS COUNTER load enable PE and the address is counted to 27. GONEXT is again low in state 27 but MIN is also low so LOADM is again low and the state will advance from 27 to 30. If in state 26 both FIXNEXT and NXTLONG are low, LOADM is high and the state machine will branch back from state 26 to one of the first states of a sequence.

States 27 and 30 are fix states which cause a FIX the beam position command to be passed to the Output Controller. The



OUTPUT STATES	CONTROL (OCTAL)	OUTCTL ROM ADDRESS A4 A3 A2 A1 A0	PROM A20 B7 B6 B5 B4 B3 B2 B1 B0	PROM A19 B7 B6 B5 B4 B3 B2 B1 B0
			DRAW, WAITENDCOUNT, RUNCOUNTER, WAITLONGDELAY	LENPENSTA, 72.2 "forms" SETANALOG, *SETSTATUS, *ENRUNEXTEND, *FORCE EXTEND, FORCE INTENSITY, DAINPUT, NXTADRMSB
NULL-MR	0	0 0 0 0 0	0 0 0 0 0 0 0 0	0 1 1 1 1 0 1 0
	1	0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
	2	0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
	3	0 0 0 1 1	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
STATUS	4	0 0 1 0 0	0 0 0 0 0 0 0 0	0 1 0 1 1 0 0 0
	5	0 0 1 0 1	0 0 0 0 0 0 1 0	0 1 1 1 1 0 0 0
	6	0 0 1 1 0	0 0 0 0 0 0 0 0	0 1 1 1 1 0 1 0
	7	0 0 1 1 1	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
FIX	10	0 1 0 0 0	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
	11	0 1 0 0 1	0 0 0 0 0 0 0 0	0 1 1 1 1 0 0 0
	12	0 1 0 1 0	0 0 0 0 0 0 0 0	0 1 1 1 1 0 1 0
	13	0 1 0 1 1	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
MOV	14	0 1 1 0 0	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
	15	0 1 1 0 1	0 0 0 0 1 1 0 0	0 1 1 1 1 0 0 0
	16	0 1 1 1 0	0 0 0 0 0 0 0 0	0 1 1 1 1 0 0 0
	17	0 0 1 1 1 1	0 0 0 0 0 0 0 0	0 1 1 1 1 0 1 0
DRAW	20	1 0 0 0 0	0 0 0 0 0 0 0 0	0 0 1 0 1 0 0 1
	21	1 0 0 0 1	0 0 0 0 0 1 0 0	0 0 1 0 1 0 0 1
	22	1 0 0 1 0	1 0 0 0 1 1 0 0	0 0 1 0 1 0 0 1
	23	1 0 0 1 1	0 0 0 0 0 0 0 0	0 0 1 0 0 0 0 1
	24	1 0 1 0 0	0 0 0 0 0 0 0 0	0 0 1 0 1 0 1 0
	25	1 0 1 0 1	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
	26	1 0 1 1 0	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
	27	1 0 1 1 1	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 0
LIGHTPEN	30	1 1 0 0 0	0 0 0 0 0 0 0 0	1 1 1 1 1 0 0 1
	31	1 1 0 0 1	0 0 0 0 0 0 0 0	0 1 1 1 1 0 1 0
	32	1 1 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	33	1 1 0 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
DOT	34	1 1 1 0 0	0 0 0 0 0 0 0 0	0 0 1 1 1 0 0 1
	35	1 1 1 0 1	0 0 0 0 1 1 0 0	0 1 1 1 1 0 0 1
	36	1 1 1 1 0	0 0 0 0 0 0 0 0	0 1 1 1 1 0 0 1
	37	1 1 1 1 1	0 0 0 0 0 0 0 0	0 1 1 1 1 1 0 0

Table 3.4

OUTCTLROM and OUTCTLREG

Address - Data Table

function performed by a FIX will be described later. In this discussion we are concerned with the signals FIXNEXT and NXTLONG either of which being high causes a dispatch through states 27 and 30 and a FIX to be sent to the Output Controller.

NXTLONG is a signal that will be high if the next vector coming down the pipe line is "declared" or defined to be long vector. (The desired result being that a FIX be applied at the end of the present vector if the next vector to be drawn is a "long vector".) Detection of a "long vector" is accomplished by examining the ENCODE bits which were selected on the 195215 ROOT CALCULATOR. Recall that these bits provide a measure of ΔX and ΔY and that the larger of ΔX and ΔY determines the quantity N used to operate the left shifter that generates $N\Delta X$. The upper three bits of ENCODE are wired to the SHIFT inputs shown on Figure 3.7. The logic for *NXTLONG indicates that it will be low, a long vector, if:

ARITHOUT . *SHIFT3 . (*SHIFT2 + *SHIFT1)

Where ARITHOUT "lets the logic" look at the SHIFT inputs. Thus a vector is long if:

ENCODE 3 SHIFT 3	ENCODE 2 SHIFT 2	ENCODE 1 SHIFT 1
0	0	0
0	0	1
0	1	0

As ENCODE (3-1) are the upper three bits of the code that specifies the number of shifts in the larger of ΔX or ΔY another interpetation of the table is that a long vector is declared when the larger of ΔX or ΔY undergoes five or less left shifts. *NXTLONG is also applied to the asychronous clear of the FIXCOUNTER. Thus it will force the output of this counter, *FIXNEXT, low.

*FIXNEXT, if low, will also cause the state machine to continue onto state 27 from state 26 rather than branch back to the first state of a sequence. It is employed when a sequence of short vectors is being generated to cause a FIX to be applied to every other short vector. Thus on successive passes through the DRAW sequence it is alternately high and low. *FIXNEXT is the least bit of the FIXCOUNTER. This counter is synchronously loaded with all ones when *SETFIXCTR is low because *FORCEFIX is tied high. It is asychonously cleared by *NXTLONG. It is allowed to count when GONEXT and MCHANGE are both high. The MULTCTLROM and MULTCTLREG Address - Data table shows that *SETFIXCTR is low and therefore that FIXCOUNTER is loaded, to all ones, in states Ø, 4, 1Ø, and 2Ø. Thus *FIXNEXT is high for all successive states of these sequences. In the DRAW sequence *SETFIXCTR is always high. The state of *FIXNEXT will be changed by counting FIXCOUNTER upon leaving state 26 because GONEXT is high. Thus on successive passes through the DRAW sequence states 27 and 30 are passed through then branched around. This is the pattern that

causes the FIX to be applied to everyother short vector in a sequence of short vectors. Notice that FIXCOUNTER is cleared by *NXTLONG to insure that a FIX is applied before a long vector is drawn. Thus a FIX is applied to everyother short in a sequence of short vectors and before any long vector is drawn.

The function of some of the key control signals is summerized below:

MCY1-MCYØ - The initial count bits that preset the MCYCLECOUNT counter. They determine the number of clock periods that the Multiplier Controller remains in a particular state.

*ROMSDTO - MULTCTLROM stored bit that is "saved" in bit Q2 of MCYCLECOUNT counter. Becomes *SETFIXCTR at the output of MCYCLECOUNT.

*LOADMULTREG - Enables the X REGISTER on the 195214 XY MULTIPLIER card. This is the register that stores the multiplier X output until the Y output is ready to be applied to the D to A Converter. See Line Generator Block Diagram figure 2.3 and 195214 logic, sheet 4.

*ROMLMR - MULTCTLROM stored bit that is "saved" in bit Q3 of MCYCLECOUNT counter. Becomes *LOADMULTREG at the output of MCYCLECOUNT.

*SETFIXCTR - When low allows the FIX COUNTER to be loaded with all ones.

*FIXNEXT - When low causes the FIX to be applied to the next vector by causing state 26 to branch to 27 then 30 of the DRAW sequence.

MOPN1 - MOPN2 - The three upper bits for the MULT-CTLROM passed to the Multiplier Controller by the Arithmetic Controller. Selected as the address when *LOADM is low. MOPN1 and MOPN2 also preset the upper two bits of the MULTIPLIER ADDRESS COUNTER.

Q3 - QØ - The outputs of the MULTIPLIER ADDRESS COUNTER which are selected as the address inputs MADR3, MADR2, MADR1 and MADRØ to the MULTCTLROM when *LOADM is high.

MNXTADRMSB - A MULTCTLROM output which is loaded into the MULTCTLREG and then selected as the most significant address bit, MADR4, to MULTCTLROM when *LOADM is high.

*DYTOXYBUS - When low enables the 13 bits of the Y LENGTH REGISTER (see Fig. 2.3 Line Generator Block Diagram) onto the tri-state XYOUT bus. It puts NΔY onto the XYOUT bus. It is connected to the tri-state enable *ENXYOUT of the 25S18's in positions 13, 15, 17 of the 195217, sheet 5. The Y, 195217 is in slot 40.

*DXTOXYBUT - Performs the same function as *DYTOXYBUS for the X LENGTH REGISTER which is on the 195217 in slot 39.

*YPOSTOXYBUS - When low enables the 12 bits of Y POSITION REGISTER onto the XYOUT bus. (see Figure 2.3 Line Generator Block Diagram). It puts Y POSITION onto the XYOUT bus. It is connected to the tri-state enable, *ENPOSOUT, of the 25S18's in positions 12, 14 and 16 of the 195217, sheet 5. The Y, 195217 is in slot 40.

*XPOSTOXYBUS - Performs the same function as *YPOSTOXYBUS for the X POSITION REGISTER which is on the 195217 in slot 39.

MULTIPLY - Goes to the select input of the 2 to 1 multiplexer that selects the multiplier outputs or the position output for the X and Y D/A converters. This multiplexer is shown on the Line Generator Block Diagram and the 195124 logic, sheet 4.

OPN2 - OPNØ - The MULTCTLREG bits that are passed forward to the Output Controller.

SETMULTOUT - "AND"ed with MALLONES to "J" the MULTOUT flip flop.

*MULTOUT - Flip flop output that is low when the Multiplier Controller is ready to pass OPN2 - OPNØ to the Output Controller.

*FORCEFIX - The T Enable and the least bit loaded into the FIXCOUNTER. Tied high through a pull up resistor. In future systems will be tied to a gated signal, via the backpanel, when a Light Pen is used with PS-2.

*NXTLONG - Indicates that the next vector to be drawn is a long vector.

MIN - MULTCTLROM stored bit which in high for the last state of each sequence of states; ØØ, Ø5, 11, 15, 22, and 3Ø. Must be high to produce a LOADM which selects the MOPN inputs, those from the Arithmetic Controller, as the MULTCTLROM address.

GONEXT - Output of MULTCTLREG. High only in state 26. Used in determining if DRAW sequence out of state 26 is to 27 or back to first state of a new sequence.

The reminder of the functions implemented on the 195216 Arithmetic card are described in the detailed card description, Section 4.6 below.

3.5 The Output Control State Machine (195213)

The Output Control State Machine is implemented on the 195213 card. It generates the control signals required to operate the X-Y Analog Cards, 195212 and also supplies control signals to the 195215 Root Calculator, 195216 Arithmetic Control, and 195211 Scope Driver.

The following figures, drawings and tables are referenced in the following description:

a.	Figure 3.9	-	Output Control State Machine Block Diagram.
b.	195213-600	-	Output Control Ligic diagrams
c.	195213-950	-	DA Output Control state diagram
d.	Figure 3.10	-	Detailed State Diagram - Output Controller
3.	Table 3.4	-	OUTCTLROM and OUTCTLREG Address-Data Table

The implementation of the Output Control State Machine is similar in structure to the Input, Arithmetic and Multiplier Controllers that were described previously.

The OUTCTLPROM, Figure 3.9, contains the data bits that participate in the generation of the control signals. Table 3.4 is a map of this PROM and the OUTCTLREG into which the PROM outputs are loaded. The state numbers, which appear in the upper left corner of the state blocks in the state diagrams,

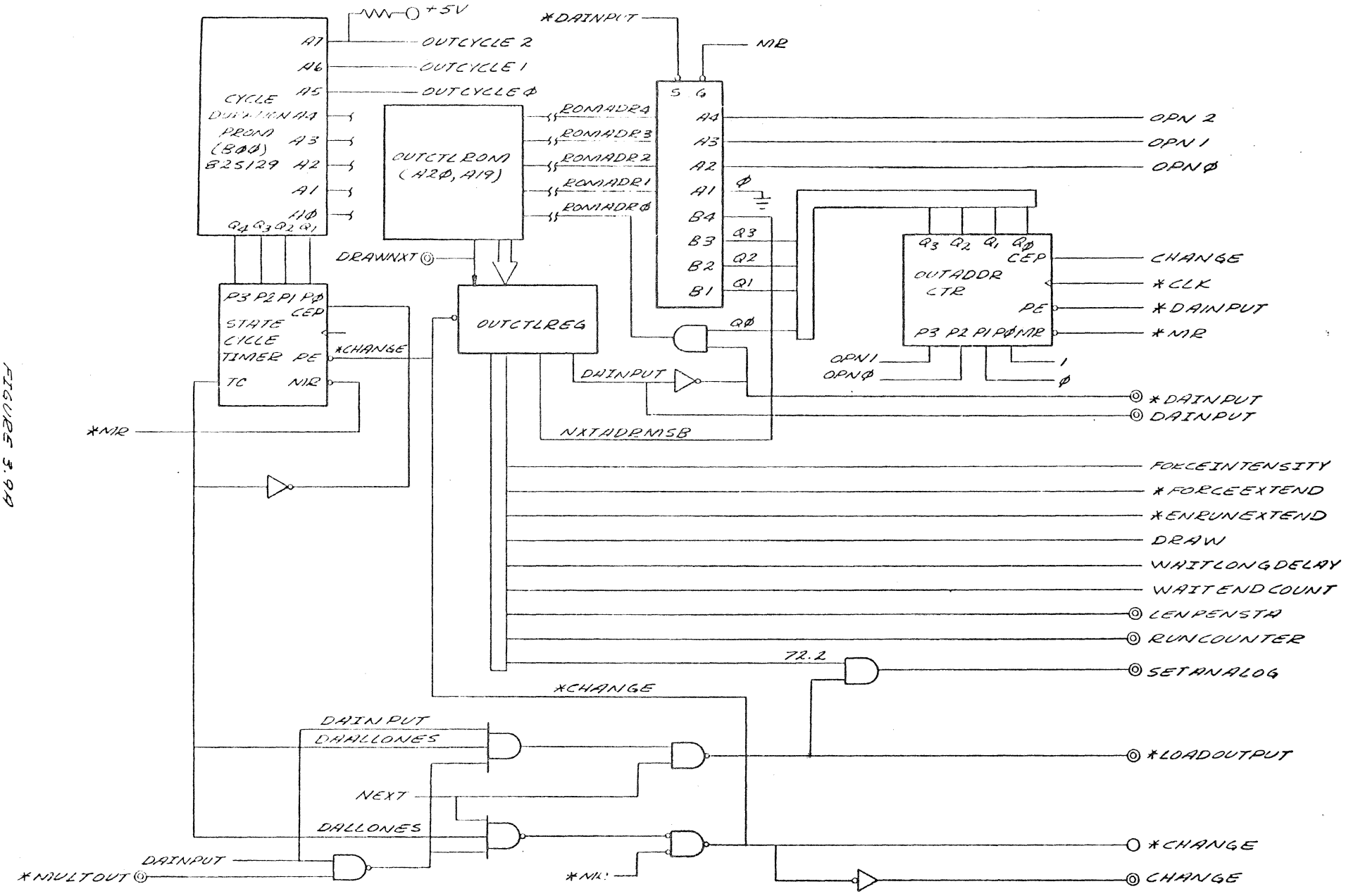


FIGURE 3.9A
OUTPUT CONTROL STATE MACHINE BLOCK DIAGRAM
(REF 195213)

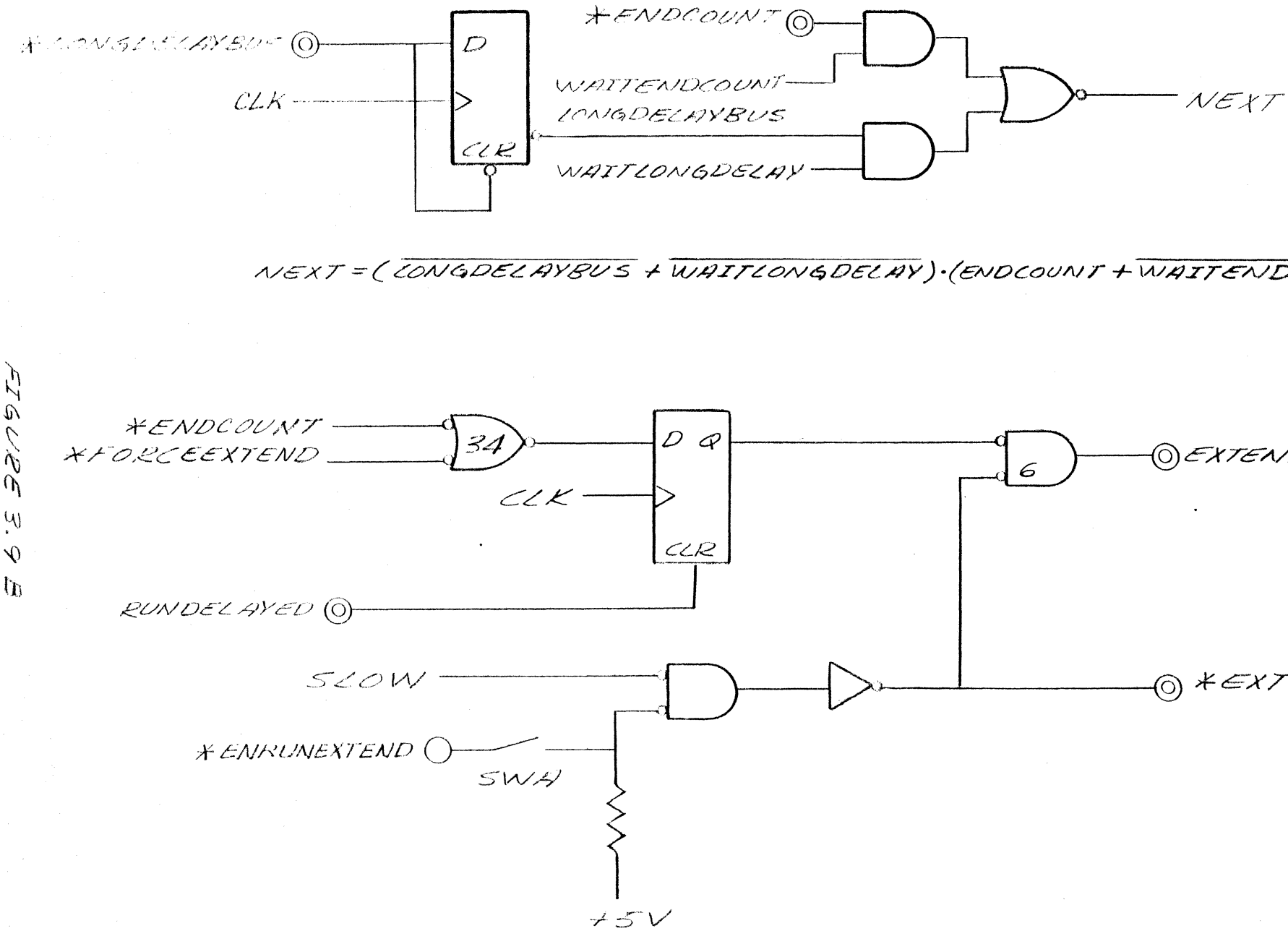


FIGURE 3.9 B
OUTPUT CONTROL STATE MACHINE BLOCK DIAGRAM
(REF 195213)

195213-950 drawing, and Figure 3.10, bear a one to one relationship to the OUTCTLPROM address inputs ROMADR4 - ROMADRØ. These address bits are applied through a 2 to 1 multiplexer. When *DAINPUT is low the A inputs OPN2 - OPNØ, Ø and *DAINPUT are applied as the five input address bits. OPN2 - OPNØ are the address inputs passed forward by the Multiplier Controller. When *DAINPUT is high the five input address bits are NXTADRMSB and Q3 - QØ. NXTADRMSB is an OUTCTLROM stored bit and Q3 - QØ are supplied by the OUTADR CTR counter. OUTADR CTR is preloaded with the count OPN1, OPN1, Ø, 1 when *DAINPUT goes low. As can be seen from the state diagram sequential states of a sequence have sequential state numbers. The state diagram shows, by the number in the lower right corner of the state blocks, that many states have rather long durations in terms of clock periods. These time periods are determined by the STATE CYCLE TIMER and the CYCLE DURATION PROM. The CYCLE DURATION PROM is addressed by the ROMADR address bits and, its upper three address bits, are OUTCYCLE2 - OUTCYCLE1. The OUTCYCLE bits are obtained from the speed control logic which is shown on sheet 4 of the 195213 logic drawing. The initial count loaded into the STATE CYCLE TIMER for the various states is shown in the table on sheet 2 of the 195213 logic.

The State Change signal *CHANGE enables the loading of the OUTCTLREG and the initial count of the STATE CYCLE TIMER from the CYCLE DURATION PROM and indicates a change of state of the controller. The complimentary logic equation for *CHANGE, ie

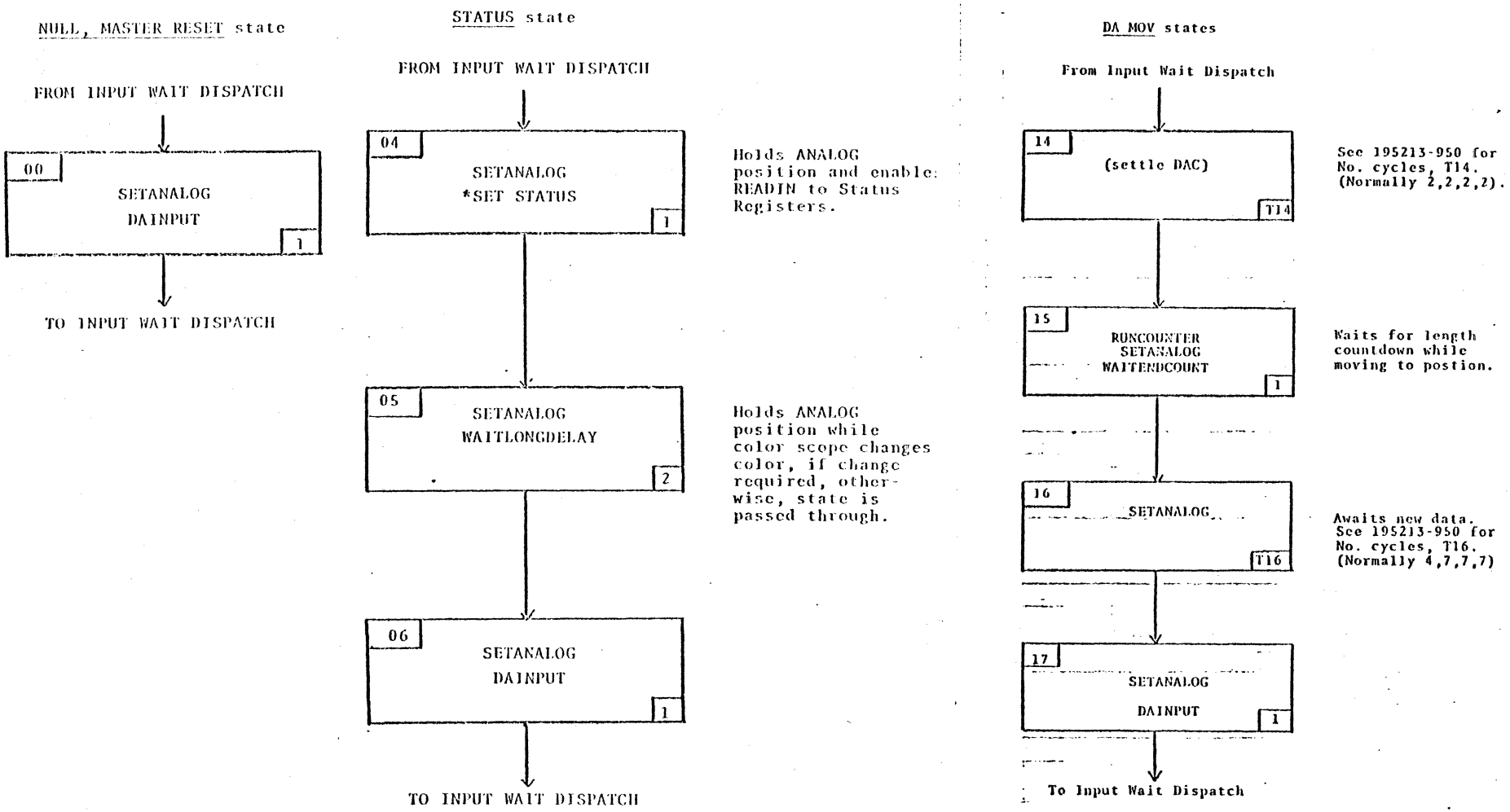


Figure 3.10

DETAILED STATE DIAGRAM OUTPUT CONTROLLER

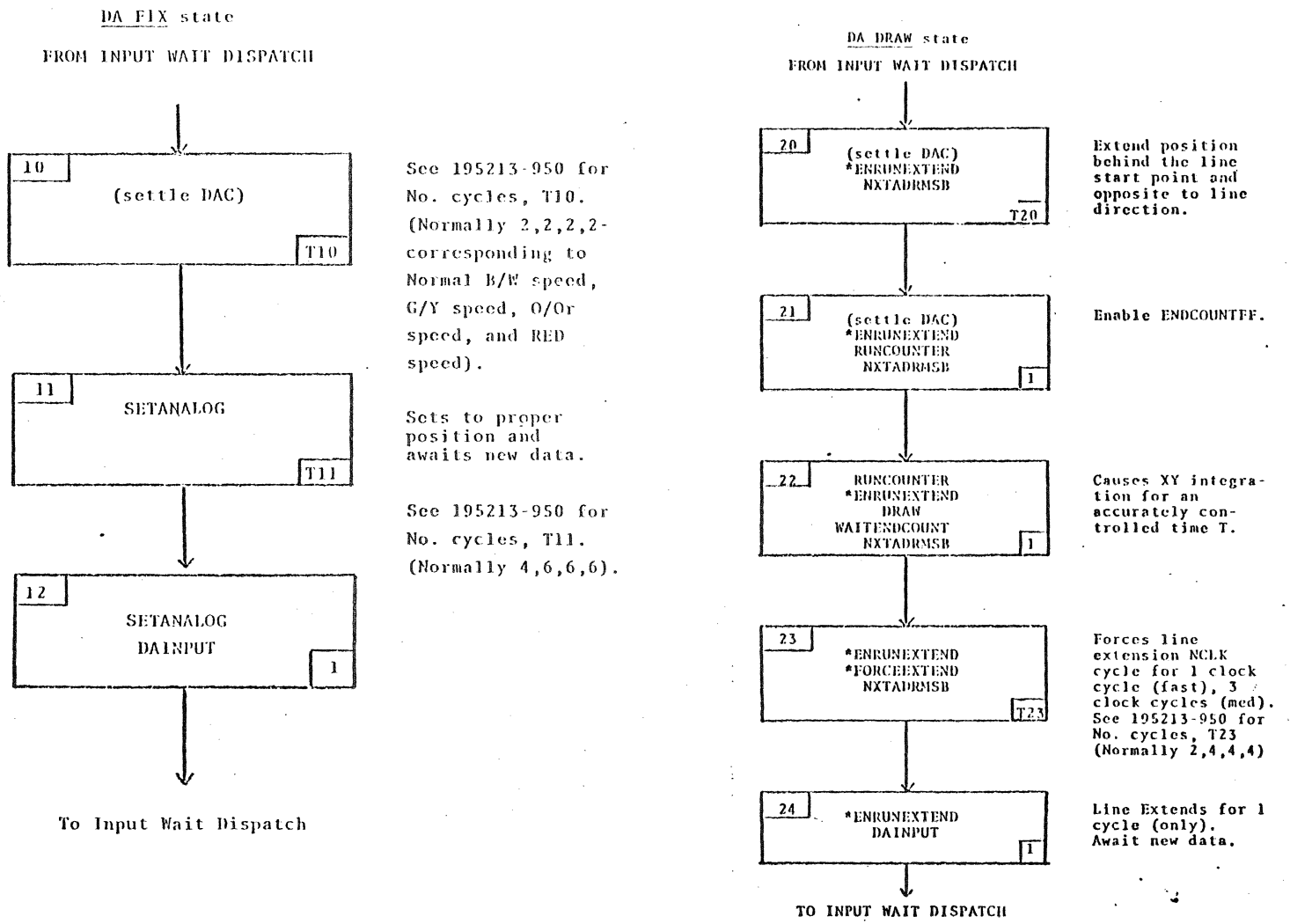


Figure 3.10

DETAILED STATE DIAGRAM OUTPUT CONTROLLER

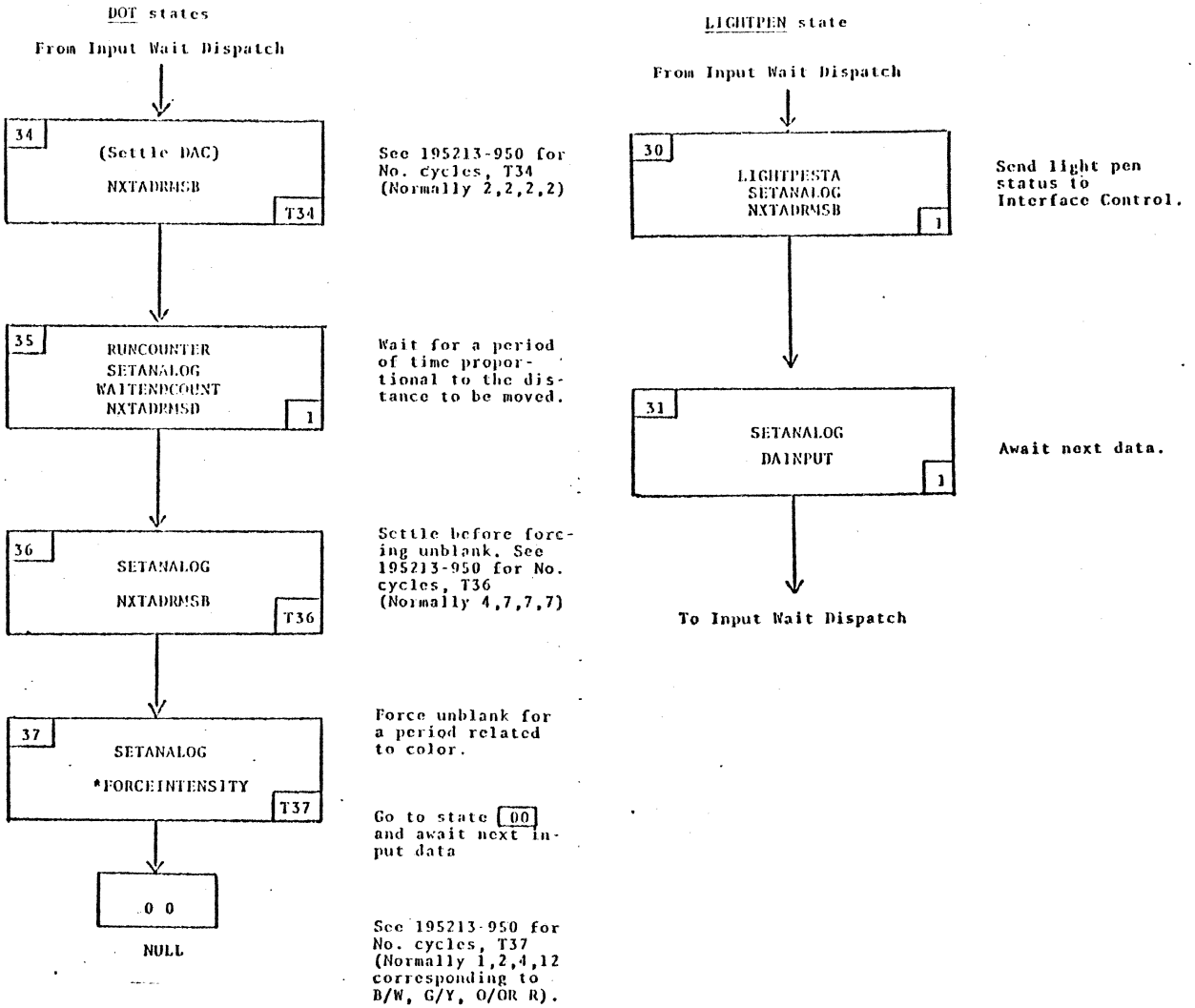


Figure 3.10

DETAILED STATE DIAGRAM OUTPUT CONTROLLER

*CHANGE is low if:

*MR + NEXT . DALLONES . (*DAINPUT + MULTOUT) where:

*MR is the L.G. reset.

DALLONES is the carry output of the STATE CYCLE TIMER indicating that the present state has existed for the desired number of clock periods.

*DAINPUT is an output of OUTPUTCTLREG and hence a bit stored in OUTPUTCTLROM. From the OUTPUTCTLROM-OUTPUTCTLREG Address. Data table it can be seen that DAINPUT is low, therefore *DAINPUT is high, for all states of a sequence except the last one of a sequence.

MULTOUT indicates that the Multiplier Controller has address bit to pass to the Output Controller and thus controls generation of *CHANGE for the last state of a sequence. The logic that generates NEXT is shown in Figure 3.9B of the block diagram. It monitors two signals of OUTCTLREG, WAITLONEDELAY and WAITENDCOUNT and the signals *LONGDELAYBUS and *ENDCOUNT. The *CHANGE equation above requires that NEXT be high to go to the next (NEXT) state. The positive logic equation for NEXT ie. NEXT is high is:

NEXT = (*LONGDELAYBUS+*WAITLONGDELAY). (ENDCOUNT+*WAITENDCOUNT)

This equation indicates that NEXT is dependent upon LONGDELAYBUS going low when WAITLONGDELAY is high and dependent upon ENDCOUNT going high when WAITENDCOUNT is high. If both WAITLONGDELAY and WAITENDCOUNT are low NEXT is high. From the OUTCTLROM and OUTCTLREG table it can be seen that the states

where NEXT controls the state change are Ø5 which uses WAITLONGDELAY and 15, 22 and 35 which use WAITENDCOUNTER. This is also shown in the corresponding state blocks of the state diagrams where the statememts WAITLONGDELAY and WAIT-ENDCOUNT appear. *LONGDELAYBUS is a delay signal which is generated by the color delay logic located on the 195211 Scope Driver card, see sheet 6. It is needed to allow for the switching time required to change the CRT anode voltage when color changes are commanded to a beam penatration color display. *ENDCOUNT is the carry output of the Integration Timer which is located on sheet 4 of the 195215 Root Calcu-lator card. This signal indicates that the time required to draw a line, a dot or to move the beam is complete. Note that the states that use WAITENDCOUNTER, 15, 22 and 35 are in the MOV, DRAW and DOT sequences.

The generation of the signals EXTENDINTGR and *EXTEND, shown in Figure 3.9B and on sheet 2 of the 195213 logic, is controlled by the OUTCTLREG outputs *FORCEEXTEND and *ENRUNEXTEND. EX-TENDINTGR and *EXTEND are used on the X-Y Analog Card, 195212 to cause the CRT beam to overshoot the final position it is being commanded to draw to. This overshoot is a compensation for the characteristics of the display deflection amplifiers and unblank circuitry and is required to obtain sharp endpoint matching in the final image. The analog circuitry effected on the 195212 card is discussed below. Our purpose here is to describe the generation of the *EXTEND and EXTENDINTGR signals.

*EXTEND will be high and EXTENDINTGR will be low if the signal SLOW is high. SLOW indicates that a line is being drawn at a slow rate and overshoot compensation is not required, it is generated by the speed control logic shown on sheet 4 of the 195213 logic. If SLOW is low, *ENTEND will be low if *ENRUN-EXTEND is low when the mechanical switch, SWA, is in the closed position. If the mechanical switch is open *EXTEND will always be high. The mechanical switch thus disables the *EX-TEND and EXTENDINTGR by forcing *EXTEND high. EXTENDINTGR can only be high if *EXTEND is low and the Q output of the flip flop is low. The Q output of the flip flop becomes low if cleared by a low RUNDELAYED or if a low output of the AND gate 34 is clocked into the flip flop. Thus the flip flop output is low when either or both *ENDCOUNT and *FORCEEXTEND are low. Assuming the machine is operated with SWA closed then:

*EXTEND is low when SLOW and *ENRUNEXTEND are both low and EX-TENDINTGR is high when SLOW and *ENRUNEXTEND are both low AND either or both *ENDCOUNT and *FORCEEXTEND are low. The OUTCTL-ROM and OUTCTLREG table shows that *FORCEEXTEND is low only in state 23 and that *ENRUNEXTEND is also low so that in this state *EXTEND is low and EXTENDINTGR is high if a slow vector is not being drawn ie. SLOW is low. The table also shows that *ENRUN-EXTEND is low for all the DRAW states 2Ø through 24 thus *EX-TEND is also low for these states. ENTENDINTGR will go high in state 23 or on the clock pulse after the Integration Timer runs out. *ENDCOUNT the input to gate 34, goes low when the Integration Timer, located on the 195215 Root Calculator, see

sheet 4, runs out. The signal RUNDELAYED, which when low clears the flip flop, is also generated on the 195215 Root Calculator, see sheet 4, and is a delayed version of *ENDCOUNT.

The signal *LOADOUTPUT enables the loading of the X and Y, D to A, input registers on the 195212, X-Y Analog Cards (sheet 1), enables the loading of the Integration Timer on the 195215 Root Calculator (sheet 4) and enables the loading of the Z DEPTH register on the 195231 Z Depth Que card, (sheet 4). The logical equation for *LOADOUTPUT low is:

$$DAINPUT \cdot NEXT \cdot DALLONES \cdot (\overline{DAINPUT} + MULTOUT)$$

This equation describes the implementation as shown. Note that $\overline{DAINPUT}$ in the last minterm is not necessary to perform the function but appears due to the manner in which the available logic elements were used. Also notice that this function is identical to the second part of the *CHANGE function described previously.

The signal SETANALOG is formed by ANDING the output bit of OUTCTLREG from pin 72.2 with *LOADOUTPUT. From the OUTCTLROM and OUTCTLREG Address-Data table it can be seen that SETANALOG will be low for all the states of the DRAW sequence, 2Ø through 24 and for the first state of DOT, 34. For the remainder of the states SETANALOG be the same as *LOADOUTPUT which is high for all those states for which 72.2 is high. This is also shown by the state diagram. SETANALOG is applied as

the SET input to the X-Y Analog card, see sheet 4 of the 195212 input connector pin 39. SET is inverted and determines the magnitude of the current supplied at the input to the X and Y deflection amplifiers when SET ANALOG is high.

There are five additional signals derived from the OUTCTLREG. Their functions are:

DRAW(DRAWNXT) - Is high during state 22 of the DRAW sequence. The output DRAWNXT from the OUTCTLROM is sent to the Root Calculator 195215 and stored in the DRAW flip flop when *CHANGE goes low.

FORCEINTENSITY - Is high during state 37 of the DOT sequence. Applied as the high order select bit to an 8 to 1 selector. It forces the selector output low and generates the UNBLANK to draw a DOT on the display. See sheet 5 of the 195213 logic, input A2 of the 82S3Ø in position 61.

LENPENSTA - Not connected on back panel. Will be used in systems which contain a Light Pen.

RUNCOUNTER - Enables the Integration Timer on the Root Calculator, 195215 sheet 4. It is high for MOV state 15, DRAW states 21, 22 and DOT state 35.

*SETSTATUS - Clocks the STATUS register on the Arithmetic Control card 195216, sheet 4. Connected to state change logic input *ENSTATUS, input connector pin 80, on the Input Control card 195218, sheet 4. Clocks the SELECT A and SELECT B inputs into a register on the Scope Driver 195211, sheet 3.

The reminder of the logic on the 195213 card is described in the card description section below.

3.6 State Machine Controller Intercommunication (195216)

In the above discussions of the State Machine Controllers the method of passing control PROM address bits forward to a controller from the previous controller was described. The function of these address bits is to select the initial state of a sequence of states which the controller steps through to perform a particular operation. When a controller is in a state where it generates the control PROM address bits for the next controller and has them on the output lines it sets an output indicator flip flop. These controllers and their output indicating flip flops are:

Input Controller	-	DATAOUT
Arithmetic Controller	-	ARITHOUT
Multiplier Controller	-	MULTOUT

The Output Controller is the last in the sequence and as it does not pass address bits to another controller it does not have an output indicating flip flop.

In order to accept the incoming address bits a controller must be in the last state of a sequence of states and must have its control PROM input select multiplexer conditioned to apply these address bits to its control PROM. The controllers and the signals that indicate that they are ready for the address inputs are:

Arithmetic Controller	-	ARITHINPUT
Multiplier Controller	-	LOADM
Output Controller	-	DAINPUT

The Input Controller receives its address bits from the FIFO, its input operation is not covered in this discussion.

The logic that is employed to regulate or time the passing of address bits from one controller to the next is shown in Figure 3.11, STATE MACHINE CONTROLLER INTERCOMMUNICATION. The decision logic is implemented in a 74S182 Look - Ahead Carry Generator. The outputs of the carry generator DATAGO, ARITHGO and MULTGO are fed back respectively to the Input Controller, Arithmetic Controller and the Multiplier Controller. Figure 3.11 shows how these "GO" signals must be high in order for the three controllers to produce their low state change signals; *CHANGEDATACTL, *ACHANGE and *MCHANGE. The inputs to the carry generator are the three output indicator flip flops and the three input ready signals described above. An additional input is applied to the carry generator, *FREEZE. This signal is employed to selectively stop the Line Generator, freeze the controllers, when a light pen is used with PS-2.

The logical equations for the "GO" signals are more easily understood if they are expressed for the low condition of the "GO" signals. That is when they are inhibiting their respective state change signals from going low. The logical equations for the low state of the "GO" signals and a description

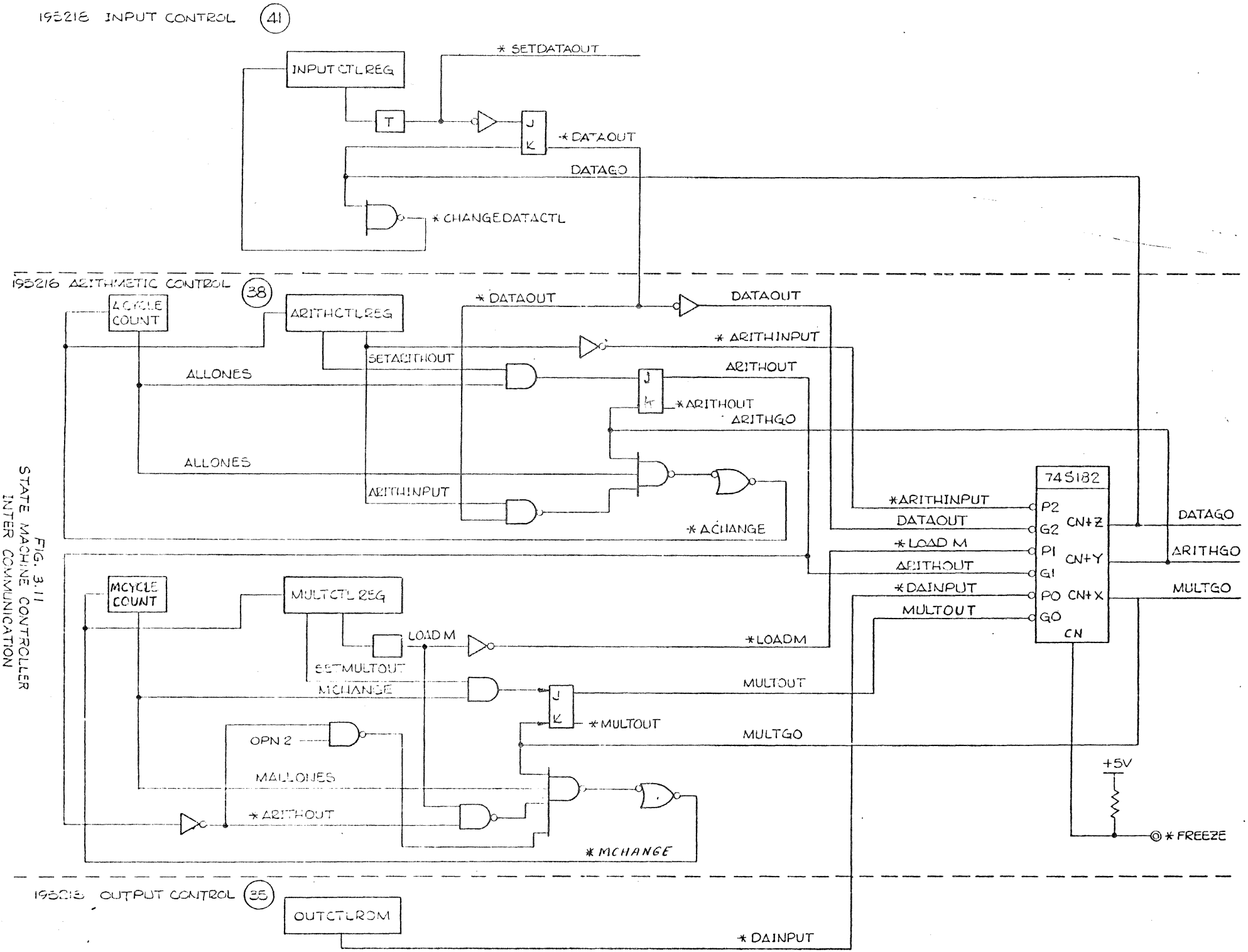


FIG. 3.11
STATE MACHINE CONTROLLER
INTER COMMUNICATION

of their meaning is as follows:

The Multiplier GO

*MULTGO = MULTOUT (*DAINPUT + FREEZE)

MULTGO is low and inhibits a *MCHANGE when the Multiplier Controller has its address output on the lines but the Output Controller is not yet in the last state of one of its state sequences or when the Line Generator is commanded to freeze.

The Arithmetic GO

*ARITHGO = ARITHOUT.*LOADM + ARITHOUT.MULTOUT
(*DAINPUT.*FREEZE + FREEZE)

ARITHGO is low and inhibits a *ACHANGE when the Arithmetic Controller has an output on the lines and any of the following conditions exist:

a. The Multiplier Controller is not in the last state of a sequence of states.

b. The Multiplier Controller also has an output on its lines and either:

   i) The Output Controller is not in the last state of one of its sequences of states

   ii) or the Line Generator is commanded to "freeze".

The Input Controller GO

*DATAGO = DATAOUT.*ARITHINPUT + DATAOUT.ARITHOUT.*LOADM + DATAOUT.ARITHOUT.MULTOUT (FREEZE + *FREEZE.*DAINPUT)

DATAGO is low and inhibits a *CHANGEDATACTL when the Output Controller has its address output on the lines but the Arithmetic Controller is not yet in the last state of a sequence of states.

OR when both the Input and the Arithmetic Controllers have address outputs on their lines and the Multiplier Controller is not yet in a last state of one of its state sequences.

OR when Input, Arithmetic and Multiplier Controllers have address outputs on their lines and either of the following conditions exist:

a. The Output Controller is not in one of the last states of a state sequence.

b. The Line Generator has been commanded to "freeze".

4.0 DETAILED DESCRIPTIONS OF THE LINE GENERATOR CARDS

4.1 195211 Scope Driver Card Description

Functions Performed

a) Sums X (Y) Deflection and X (Y) Extend voltages and supplies summed output as X (Y) display deflection inputs.

b) Provides tapped delay line circuit for modifying the display unblank pulse to match the display being driven

c) Accepts the seven bit digital Z axis intensity inputs and generates the video input for the display.

References

a) Family of 195211 logic drawings which include several versions of this card as designated by the following dash numbers:

-600 for Two Xytron or Two Color Displays
-601 for One Xytron or One Color Display
-602 for Two Xytron Displays
-603 for One Xytron Display

b) The description of the operation of the scope

driver is explained for the -103 version of the card. The difference between the -103 and the other versions are discussed at the end of this card description.

X and Y Deflection (sheet 1)

The X (Y) deflection amplifiers are summing amplifiers which add the deflection and extend voltages, generated on the 195212 Analog card to produce the composite deflection signals sent to the display.

The deflection and extend voltages are summed by an input resistor network/divider whose output is applied to the non-inverting input of a LF356H operational amplifier. The return legs of the deflection and extend inputs are applied through a similar summing network to the inverting input of the operational amplifier.

Display Unblank (sheet 2)

The XY-tron display beam is unblanked, turned on, when a level of +3 volts or higher is applied to its UNBLANK input.

The input to the unblank logic is the signal * SCOPE DRIVER BLANK which is generated on the Output Control card, 195213, (sheet 5). When this signal is high the display is to be unblanked. * SCOPE DRIVER BLANK is input to a 250 n second,

tapped delay line. The two ten position switches S2 (position 35) and S1 (position 44) and the dual NPN transistor, Q2T2222, in position 37 are provided so that an UNBLANK pulse of the proper time relationship and pulse width can be generated from the * SCOPE DRIVER BLANK pulse. Modification of the time at which the leading edge occurs and the width of the display unblank input required varies from display to display due to the inherent bandwidth characteristics of various displays.

It is important that the reader understand how the blanking pulse timing and pulse width modification is performed by this circuit as its operation is key to obtaining a proper end point match.

The waveform sketches of figure 4.1-1 illustrate how the leading edge and pulse width of the * SCOPE DRIVER BLANK pulse is modified by the circuit. The sketches are drawn for a 3 μ sec, 3000 n sec, * SCOPE DRIVER BLANK input pulse with switch S2 set to position 7 and switch S1 set to position 4. The OR output is sketched at pins 3 and 5 of Q2T2222, the NPN transistor in position 37. The S2 output is the input delayed by 162.5 n sec. The S1 output is the input delayed by 200 n sec. The OR output is a pulse delayed by 162.5 n sec having a width of 3037.5 n sec. The additional 37.5 n sec are obtained by ORing the S2 and S1 outputs which are replica's of * SCOPE DRIVER UNBLANK separated from each other by 200 - 162.5 = 37.5 n sec.

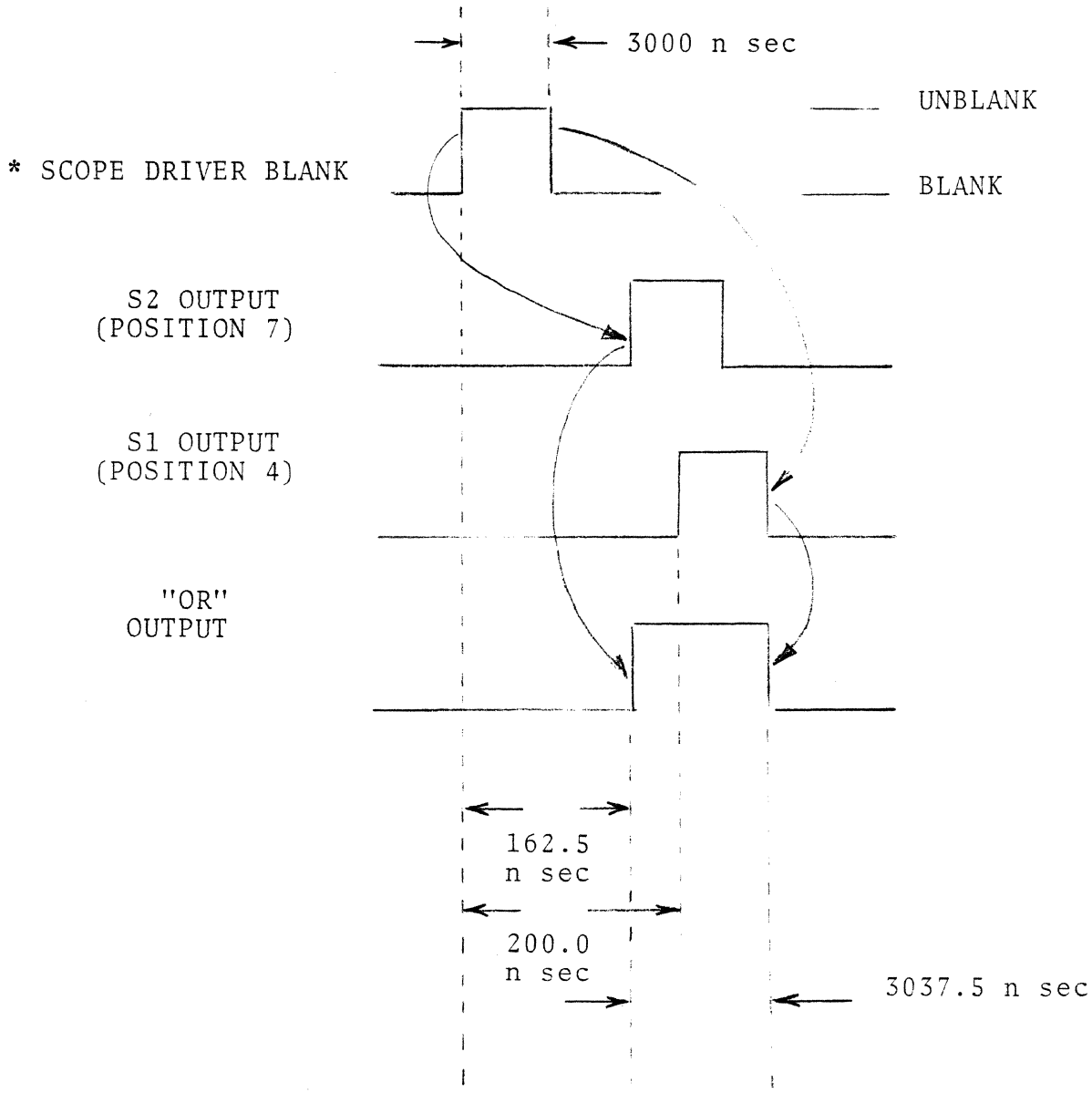


Figure 4.1-1 UNBLANK Pulse Formation

It is extremely important to be aware of how the select switches S2 and S1 are connected to the delay line. These connections are summarized by table 4.1-1.

SWITCH POSITION	DELAY LINE TAP (n sec)	
	S2 (POS 35)	S1 (POS 44)
10	112.5	125.0
9	137.5	137.5
8	150.0	150.0
7	162.5	162.5
6	175.0	175.0
5	187.5	187.5
4	200.0	200.0
3	212.5	212.5
2	225.0	225.0
1	250.0	237.5

TABLE 4.1-1 UNBLANK Tap Switches

Several important features are revealed by this table:

a) S2 and S1 positions 9 through 2 are connected to the same delay line tap. Therefore if both are set to the same position, 9 through 2, the pulse will be delayed but its width will be the same as *SCOPE DRIVER BLANK

b) The general approach is to use S2 to determine the pulse delay time and S1 to determine the

pulse width. Thus in general S2 will have a higher position on than does S1. However notice that if S2 is set to position 4 and S1 to position 7, the reverse of the example above; the end result is the same.

c) Position 10 of S2 is connected to the 112.5 tap and position 10 of S1 to the 125.0 tap. Thus if S2 has position 10 on it will determine the time of the leading edge. i.e. S1 will have no effect on the leading edge if S2 has position 10 on.

d) Position 1 of S2 is connected to the 250.0 tap and position 1 of S1 to the 237.5 tap. Hence if S2 has position 1 on, S1 determines the leading edge.

The recommended approach for end point matching is to use S2 to determine the leading edge and S1 the pulse width. Using this approach the upper limit that can be obtained is a 250.0 n sec delay, S2 position 1, and no change in pulse length, S1 all open. To obtain a 237.5 n sec delay S1 is closed and S2 closed to obtain a 12.5 n sec increase in pulse width or S2 open for no increase in pulse width.

The output of the OR gate is applied to one input of the three input AND in position 21. One of the other inputs is ENA

which is the buffered display select signal SELECT A. SELECT A is YCBUS 10 and corresponds to STATUS command, word 2, bit 10, the scope 1 enable. SELECT A is clocked into the 74174 buffer by the inversion of * SETSTATUS which is an output of the OUTCTLREG of the 195213, sheet 1. The third input is the inversion of the SYSTEM RUNNING multivibrator of the 195213, sheet 5. The output of the AND is applied to a 2N2219 emitter follower which drives the coaxial cable connected to the display UNBLANK.

Video (sheet 3 and 4)

The logic for the video, the Z axis intensity, is shown on these sheets. At the left on sheet 3 the seven bits of Z are input from the Z Depth Que card, 195231. The seven Z inputs and the inputs ACONTRAST (1,Ø) address the intensity function PROMS, AØ2 and AØ3, in positions 16 and 31. The input ACONTRAST 2 is applied to the enable input of the PROMS. The inputs ACONTRAST (2 - Ø) are wired to the five position contrast select switch on the display via the 195243 cable assembly. Table 4.1-2 shows the levels of the ACONTRAST signals as a function of contrast switch position. The actual wiring of the Display Contrast Switch is shown in Figure 4.1-2.

DISPLAY CONTRAST SWITCH POSITION	ACONTRAST Ø	1	2
1	H	H	H
2	L	L	L
3	H	L	L
4	L	H	L
5	H	H	L

TABLE 4.1-2 Contrast Switch Select

On early ECO levels, A2 and below, the four output bits of the two intensity function PROMS are applied to 25S1Ø four bit shifters. The select signals GAIN Ø and GAIN 1 are generated on the 195213 Output Control card sheet 4; ZGAIN Ø and ZGAIN 1.

On later versions A3 and above the shift inputs are grounded and the shifters do not shift the input data, which are merely passed through the 25S1Ø's and input to the 74S374 buffer in position 33. The shift operation is replaced by a multiplication. This multiplier is shown on sheet 4 of the -103, A3 logic and is accomplished by modifying the positive reference voltage applied to the D/A converter. The signals GAIN Ø and GAIN 1 drive a resistor voltage divider

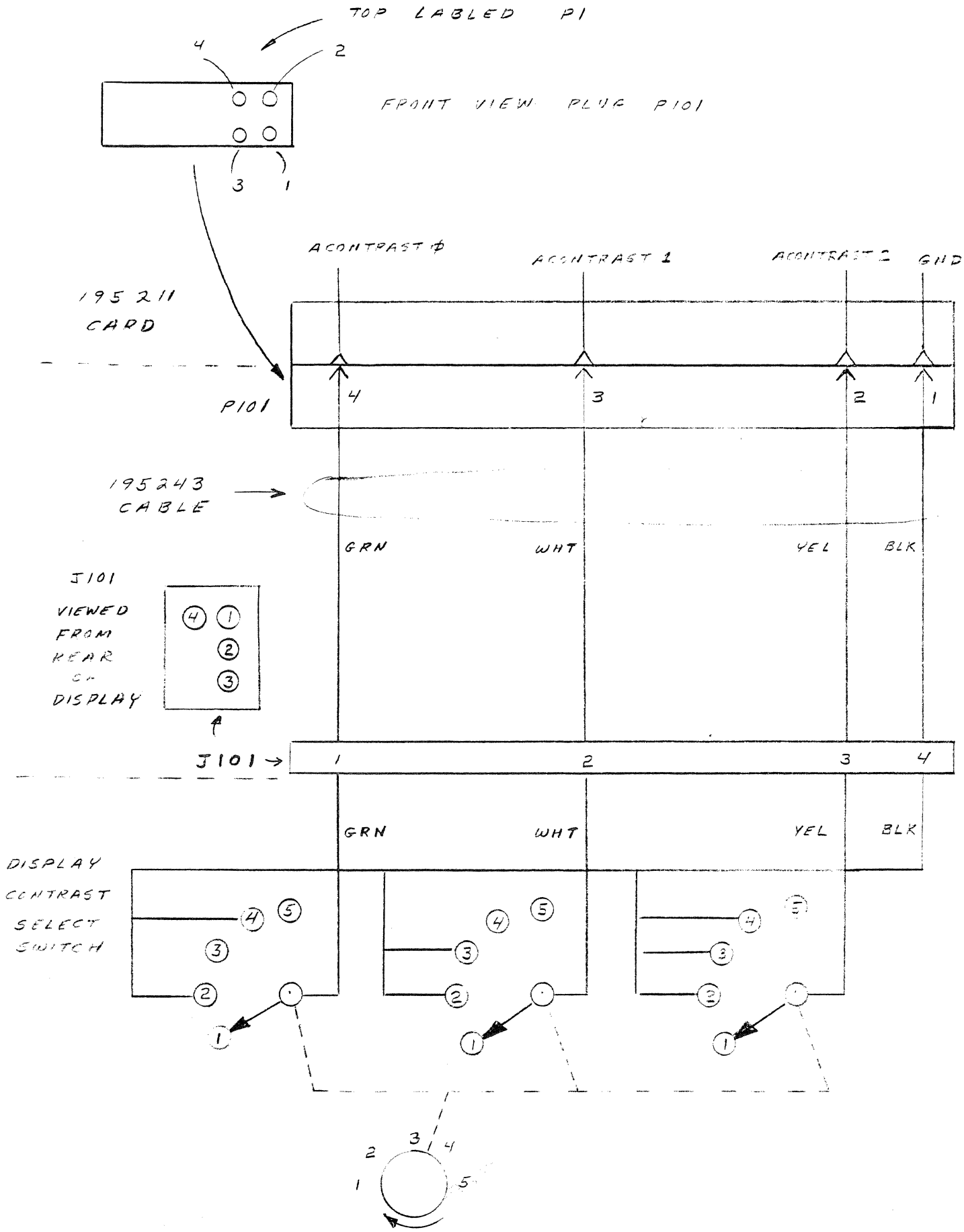


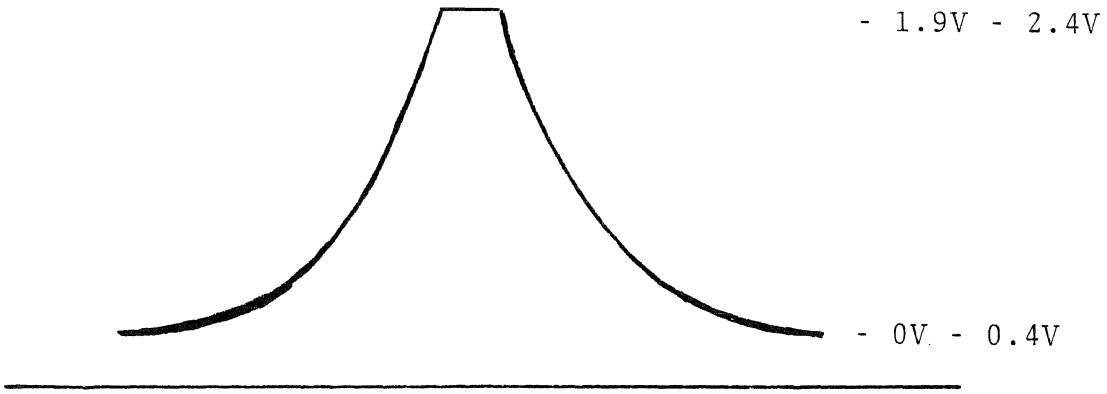
Figure 4.1-2 Contrast Switch Wiring

which determines the value of DACREF applied to the positive reference voltage input of the DAC. The voltage, DACREF, as a function of the levels of GAIN 1 and GAIN Ø are:

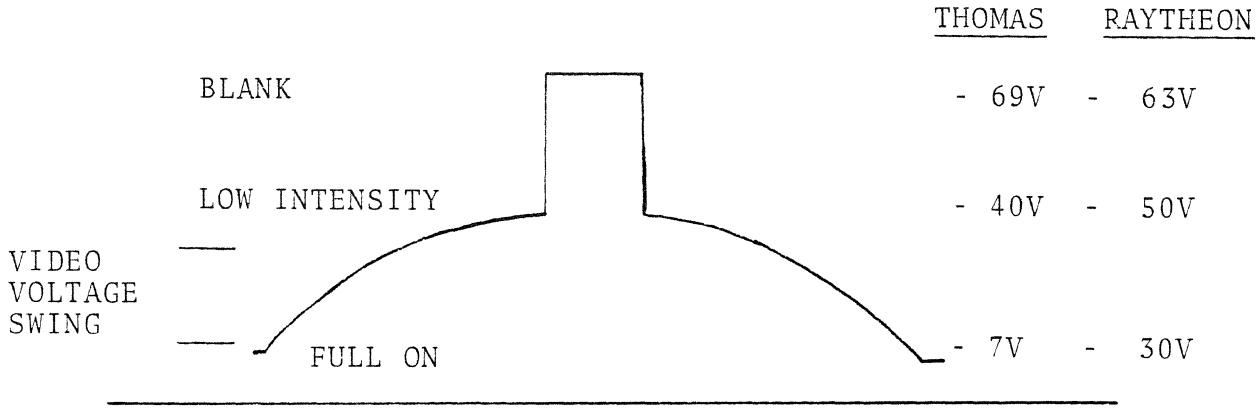
GAIN 1	GAIN Ø	DACREF (volts)
L	L	15.0
L	H	7.9
H	L	10.8
H	H	5.8

On still higher ECO levels of the 195211 the shifters have been elimated entirely.

The outputs of the 74S374 latch in position 33 are applied to an 8 bit digital to analog converter, a DAC-Ø8A, position 41 whose outputs, IOUT and $\overline{\text{IOUT}}$ are applied to the differential amplifier, MSK730, in position 47. The delay line DL2, position 56, and the switch S5, position 55 allow adjustment of the delay of the video signal for matching to different displays. The buffer amplifier, sheet 4, is an MSK730, position 53. The input resistor to this amplifier consists of a 2.37K fixed resistor in series with a 5K potentiometer, R69, the waveform of the video signal at the output of the buffer amplifier, VIDEOAOUT, when the boxes diagnostic pattern is displayed and the display contrast switch is in position one is shown below.



The mixed video and unblank signal when viewed at the cathode of the display CRT should have the waveform shown below. Note that the voltage values are different for the Thomas and the Rayheon CRT's



It should be possible to approximate these video voltage swings shown in the above waveform by adjusting the current input into output amplifier with R69, sheet 4.

-100 and -101 Versions of the 195211

These are older versions of the 195211. Their operation is essentially the same as that explained for the -103. Originally the scope driver was designed to drive one and/or two displays. The -100 and -101 differ only in that the -101 contains only those components needed to drive display A. Sheets 1 and 2 show the X and Y deflection amplifiers.

Sheet 3 shows two UNBLANK circuits. The AUNBLANK and BUNBLANK outputs are selected by the AND gates in position 20 by the buffered SELECT A and SELECT B inputs which are off YCBUS 10 and 11 and correspond to STATUS command bits 10 and 11, scope 1 and scope 2. Two other control inputs, * BLINKDASH 1 and * BLINKDASH 2 are shown. These are not used: their back panel pins are unwired and they are pulled high by being connected to +5 volts by 4.7K resistors.

The delay line and the tap switches for modifying the * SCOPE DRIVER BLANK pulse differs from that of the -103 version. * SCOPE DIRVER BLANK is inverted by the 74S37 in position 10 before the delay line however the nand gate in position 28 which implements the OR function results in a re-inversion of the unblank pulse. The major difference is in how the switches are connected to the delay line taps. The switches are 8 position and are connected as shown in the following table:

SWITCH POSITION	DELAY LINE TAP (nsec)	
	S1 (53)	S2 (54)
8	150.0	162.5
7	162.5	175.0
6	175.0	187.5
5	187.5	200.0
4	200.0	212.5
3	212.5	225.0
2	225.0	237.5
1	237.5	250.0

Notice that switch S1 (53) is the one to be used to determine the total delay of the unblank pulse and S2 (54) to modify the pulse width. (This differs from the -103 version where S2 is used to determine total delay and S1 the pulse width. Another difference is that no S1 (53) switch position is connected to the same tap as the corresponding S2 (54) switch position).

Sheets 4 and 5 show the Video or Z axis intensity logic for displays A and B and sheet 8 shows the Video output amplifiers for both A and B displays. As explained above early versions of the 195211 used 25S1Ø shifters to shift the data output by the intensity function PROMS, AØ3 and AØ2. The number of shifts being determined by the signals GAIN 1 and GAIN Ø. These signals are generated on the 195213, Output Control card, see ZGAIN 1 and ZGAIN 2 sheet 4.

Sheet 8 shows the output video amplifiers. The gain of these amplifiers are adjusted by R69 and R76 which vary the input current to the operational amplifier. R73 and R80 vary the output offset voltage and thus can be used to set the maximum display brightness level.

Sheets 6 and 7 contain logic for decoding color codes, generating anode switching signals for beam penatration CRT's and modifying the vector drawing time as a function of color

to be displayed. As color CRT's are not supported by the -100 and -101 cards this logic is not used.

-102 Version of the 195211

The -102 is a dual display version of the -103 and is functionally the same. It contains sufficient components to drive two displays.

4.2 195214 X-Y MULTIPLIER Card

Function Performed

a) Shifts the normalized quantities NΔX and NΔY as a function of the length of the vector and the color of the vector to be drawn.

b) Performs the multiplications required to compute the sine and cosine inputs to the integrators.

$$\text{COS } \Theta = \frac{N\Delta X}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

$$\text{SIN } \Theta = \frac{N\Delta Y}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$$

c) Selects the X and Y position data for a MOVE or the COS Θ and SIN Θ data for a DRAW.

d) Buffers the X or COS Θ output data to the integrator while the Y or SIN Θ output data is being computed.

e) Selects the X or COS Θ and the Y or SIN Θ quantity on to the CBUS for maintenance monitoring.

References

a) Figure 2.3 Line Generator Data Flow Block Diagram (this manual).

b) 195214-660 X-Y Multiplier Logic Diagram.

c) 195214-900 X-Y Multiplier Block Diagram.

X-Y Shifter (sheet 1)

Thirteen bits of NΔX and NΔY from the X-Y Adder-Shifter, 195217, are input via the XY or XYOUT Bus. The shift select signals, SØ and S1, are generated by the CHARACTER SPEED ROM on the 195213, Output Control card, sheet 4, position 14 where their names are XYMULT Ø adn XYMULT 1. The twelve shifter output bits MU (11-Ø) and the sign, XY12, form the multiplicand.

Multiplier (sheets 2 and 3)

The multiplier for multiplying NΔX and NΔY by $\frac{1}{N\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$ to compute COS Θ and SIN Θ is shown on sheets 1 and 2. The multiplier is implemented with Advanced Micro Devices, AM25S05, four bit by two bit 2's compliment multiplier IC's. The multiplicand is the sign XY12 and the twelve outputs bits of the shifter MU (11 - Ø). The multiplier is the nine bit

inverse square root generated on the 195215 Root Calculator card, MULT (8 - Ø), sheet 2. The eleven bit product is * ALL and A (1Ø - Ø), bottom of sheet 3.

X, Y Position or COS Θ, SIN Θ Select (sheet 4)

The multiplier product *ALL, A (1Ø - Ø) and the X, Y Position quantity, direct from the XYBUS, * SIGN, XY (11 - 1) are applied to the 2 to 1 selector, positions 24, 12 and 15. The select signal, MULTIPLY, is an output of the MULTCTLREG on the 195216 Arithmetic Control card, sheet 3. When MULTIPLY is high a DRAW is being executed and the multiplier outputs COS Θ and SIN Θ are to be sent to the integrator. When MULTIPLY is low a MOVE is being executed and X and Y POSITION are sent to the integrator. The 25SØ7's in positions 17 and 14 form the X buffer register. This register holds the X data, X POSITION or COS Θ, while the Y data, Y POSITION or SIN Θ, is being calculated by the multiplier. Buffering of the X data is required so that X and Y can be simultaneously applied to the integrator, 195212 XY Analog card. The enable for the X buffer is * LOADMULTREG which is also an output of the MULTCTLREG on the 195216. Recall however that * LOAD-MULTREG actually is stored in the Q3 flip flop of MCYCLECOUNT which is used as an extension of the MULTCTLREG.

The three 74S257 selectors on the left of the sheet, positions 16, 11 and 13, allow the XOUT and YOUT, inputs to the integrator, to be placed on the CBUS for maintenance purposes.

The selector enable *READTOCBUS and the select *XTOCBUS both come from the MAINTenance PROM, position 14, on the 195128, LG Input Control card, sheet 2. They are the signals *READ-MULTTOYCBUS and SEL Ø respectively on the 195128.

4.3 195215 LINE GENERATOR ROOT CALCULATOR

Functions Performed

a) Calculates the squares of $N\Delta X$ and $N\Delta Y$ and sums them to form $(N\Delta X)^2 + (N\Delta Y)^2$.

b) Calculates the square root and inverse square root of the sum of the squares.

$$\sqrt{(N\Delta X)^2 + (N\Delta Y)^2} \quad \text{and} \quad \frac{1}{\sqrt{(N\Delta X)^2 + (N\Delta Y)^2}}$$

c) Compares XENCODE and YENCODE and selects the smaller as the normalization code N.

d) Adds the ADD SHIFT control code to the normalization code N to generate the compensation and denormalization code S = ADD SHIFT + N.

e) Employs S to denormalize the square root of the sum of the squares. ie Calculates $\sqrt{\Delta X^2 + \Delta Y^2}$ from $\sqrt{(N\Delta X)^2 + (N\Delta Y)^2}$

f) Counts $\sqrt{\Delta X^2 + \Delta Y^2}$ down to control the integration time for performing a move or draw.

References

a) Figure 2.3 — Line Generator Data Flow Block Diagram - this manual

b) 195215-600 — Line Generator Root Calculator - Logic Diagram

c) 195215-900 — Line Generator Root Calculator - Block Diagram

Square $N\Delta X$ and $N\Delta Y$ (sheet 1)

Nine bits of $N\Delta X$ and $N\Delta Y$ from the 195217 X-Y Adder Shifter card are applied as the address inputs to the "Squaring Roms" in positions 18, 17, 27 and 26 respectively. The 8 bit outputs XX7 - XXØ and YY7 - YYØ represent the most significant bits of $N\Delta X$ and $N\Delta Y$ squared with the low order bits truncated.

$(N\Delta X)^2 + (N\Delta Y)^2$ and $\frac{1}{\sqrt{(N\Delta X)^2 + (N\Delta Y)^2}}$ (sheet 2)

The eight bits of $(N\Delta X)^2$ and $(N\Delta Y)^2$, XX7 - XXØ and YY7 - YYØ are the inputs to the 8 bit adder mechanized by the 74S283's in positions 35 and 36. The output SUM (7-1) and the adder CARRY output are applied to address Inverse Root ROM's in positions 3Ø and 31 to obtain the eight bit inverse root MULT (7-Ø). The sign bit is the inverted carry out of the adder MULT 8.

Determining the Normalization Code, N (sheet 2)

The normalization decision logic which compares the ΔX and ΔY normalization codes, X ENCODE and Y ENCODE, and selects the smaller as the normalization code N that is applied to both ΔX and ΔY is shown on the left of sheet 2. Figure 4.3-1 is a detailed block diagram of this logic and shows how it is connected to the X ENCODE and Y ENCODE logic on the 195217 X-Y Adder Shifters. The two four bit ENCODE tri-state busses are XSH (3-Ø) and YSH (3-Ø). The X ENCODE and Y ENCODE quantities are applied to the 74S85 comparator in position 12. The A > B comparator output is high if Y ENCODE is greater the X ENCODE. A > B output is connected to the select input of the 2 to 1 multiplexer - register SHIFT, the 25SØ9 in position 21, and selects the smaller of X ENCODE and Y ENCODE for strobing into the register when SHIFT, connector input 67, goes high. SHIFT is the output FINDSHIFT from the Arithmetic Control Register on the 195216, sheet 1. Notice that FINDSHIFT also enables the tri-state outputs of X ENCODE and Y ENCODE onto the XSH and YSH busses from the 195217. When FINDSHIFT goes low it enables the smaller of X ENCODE and Y ENCODE onto both the XSH and YSH busses by enabling the 8T95 in position 1Ø and disables, forces to the high state, the outputs of the 74S257's on the 195217 cards. The selected normalization code N is also saved in the 25SØ7 register, DSHIFT in position 22. This register is enabled by * PDSHIFT

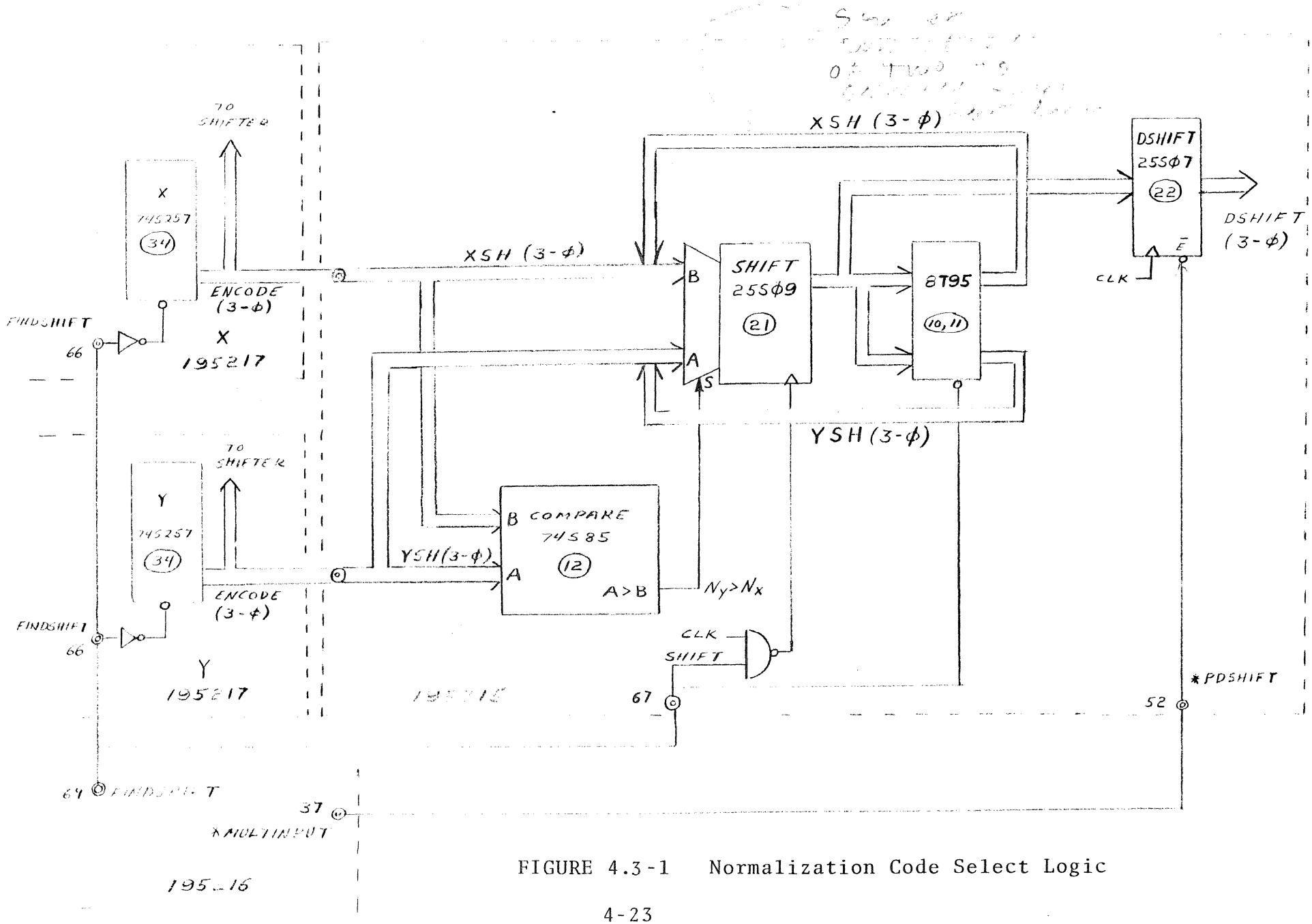


FIGURE 4.3-1 Normalization Code Select Logic

which is generated on the 195216 Arithmetic Control card, sheet 3. On the 195216 the signal is called * MULTINPUT and is low when the MULTIPLIER ADDRESS COUNTER on the 195216 is to be loaded with its initial count. (see pages 3-40, Section 3-4 above). The saved N code, DSHIFT (3-Ø) is added with ADDSHIFT (3-Ø) and used to denormalize N $\sqrt{(\Delta X)^2 + (\Delta Y)^2}$ in the integration timing logic, see below.

$\sqrt{(N\Delta Y)^2 + (N\Delta Y)^2}$ and $\sqrt{\Delta X^2 + \Delta Y^2}$ (sheet 3)

The eight bits SUM (7-1) and CARRY representing $(N\Delta X)^2 + (N\Delta Y)^2$ are applied to the address inputs to the Root ROMS in positions 32, 33, and 34. The eleven bit output *R (11-Ø) represents N $\sqrt{\Delta X^2 + \Delta Y^2}$. The normalized square root *R (11-Ø) is shifted in the two level shifter shown in the center and right of sheet 3. The 16 bit shifter output *C (15-Ø) represents the denormalized square root: $\sqrt{\Delta X^2 + \Delta Y^2}$.

The four shift control inputs S3 - SØ are the sum of DSHIFT (3-Ø) and ADDSHIFT (3-Ø) formed by the 74S283 adder position 13 shown at the upper left of sheet 3. DSHIFT is the normalize code N found by selecting the smaller of X ENCODE and Y ENCODE; the logic described above. The quantity ADDSHIFT (3-Ø) comes from the Shift Code Control Rom on the Output Control card, 195213 sheet 4.

Integration Timing (sheet 4)

The integration timing logic consists of a 12 bit counter, positions 45, 46, and 47, the delay select register, position 39, the * ENDCOUNT and DRAW flip flops, position 28, a tapped delay line with a tap select switch, positions 37 and 38, and output gating for producing the INTEGRATE and RUNDELAYED signals.

The compliment of the line (vector) length quantity $\sqrt{\Delta X^2 + \Delta Y^2}$, (the 16 bits *C (15-Ø)), are loaded into the 12 bit counter and the 4 bit delay select register when * LOADCOUNTER goes low. The * ENDCOUNT flip flop D input, * ENDGATE, goes high setting * ENDCOUNT and enabling the counter when RUNCOUNTER goes high.

The 12 bit counter is counted by the 100 n sec clock. Each count represents a 100 n sec minimum interval of integration time. As the vector being drawn approaches its end point the counter over flows and clears the * ENDCOUNT flip flop. *END-COUNT is applied to the input D7 of the select switch, 74S151, in position 38. If the vector endpoint occurs on an even multiple of 100 n sec the length bits *C (3-1) are high, thus * DLYSELECT (3-1) are high, and the switch input D7 is passed to the output W allowing INTEGRATE and RUNDELAYED to go low. If the vector endpoint is not on a multiple of 100 n sec time delay is added by selecting * ENDCOUNT from one of the taps of the delay line by the switch in position 38. The delay

line tap and amount of delay added as a function of the bits *C (3-1) (* DELAY SELECT (3-1)) is:

*DELAYSELECT 3	2	1	Ø	W OUTPUT	DELAY ADDED n sec
0	0	0		* DØ	87.5
0	0	1		* D1	75.0
0	1	0		* D2	62.5
0	1	1		* D3	50.5
1	0	0		* D4	37.5
1	0	1		* D5	25.0
1	1	0		* D6	12.5
1	1	1		* D7	0.0

Notice that the least bit *CØ (*DELAYSELECT Ø) if low inserts the additional delay of a 74SØ8 in the path, typically 7.5 n sec.

The DRAW flip flop, position 28, is set by DRAWNEXT when CHANGE is high. DRAWNEXT comes from the OUTCTLROM, on the Output Control card, 195213, sheet 1. CHANGE also comes from the 195213 and is the inverse of the signal that strobes OUTCTLROM into the OUTCTLREG. DRAW must be high when a integration is to be performed. ie. INTEGRATE to be high.

The output INTEGRATE goes to the Output Control card, 195213, (sheet 5) where it is used in generating the ANLOGINTEGRATE and *SCOPEDRIVEBLANK signals. The output RUNDELAYED also

goes to the 195213, sheet 2, where it is used to clear a flip flop that must be low, see page 3-59 above, to turn on the extend function on the ANALOG (integrator) card, 195212.

4.4 195217 X-Y Adder Shifter Card Description

Two of these cards are used. One for X and the other for Y. Respectively they are in backpanel slots 39 and 40.

Functions Performed

a. Provides storage for the eight 12 bit quantities that are used in the arithmetic calculations. i.e. XYIN, XY, XYSAVE, XYHOME, DXYDISPL, DXYSUBDIPL, XYSUB and LIMIT.

b. Performs arithmetic operations required to calculate $\Delta Y = Yn - Yo$ and $\Delta X = Xn - Xo$ for MOVE, DRAW, RELATIVE MOVE, and RELATIVE DRAW.

c. Stores Xn and Yn in the POSITION REGISTERS for use during MOVE operations.

d. Calculates the shift code, ENCODE for block normalizing ΔX and ΔY. XENCODE and YENCODE are sent to the 195215, Root Calculator, where the smaller of the two is selected and applied to both the X and Y Adder-Shifter cards as ENCODE.

e. Calculates the normalized ΔX and ΔY, $N\Delta X$ and $N\Delta Y$.

References

a. 195217-950 X-Y Adder Shifter Block Diagram.
b. 195217-600 X-Y Adder Shifter Logic Diagram.
c. 195216-900 Arithmetic Controller State Diagram.
d. Figure 2.3 Line Generator Data Flow Block Diagram.

Sheet 1

The eight word, 12 bit register file and the ALU's are shown on this sheet. The register file is implemented with 74172 Multiport Register File integrated circuits. These IC's each contain 8 words of 2 bits each which are organized into two sections.

Section 1 - Allows writing of data into any two bit word while simultaneously reading another location. The IC inputs and outputs as shown on the logic are:

write address	1W0, 1W1, 1W2
write enable	1GW
data inputs	1DA, 1DB
read address	1R0, 1R1, 1R2
read enable	1GR
data outputs	1QA, 1QB

Section 2 - Has common read/write address circuitry. Thus it has three modes of operation:

a. Writing new data into two bits
b. Reading from two bits
c. Writing into and simultaneously reading from the same two bits

The inputs and outputs as shown on the logic are:

write/read address	2W/R0, 2W/R1, 2W/R2
write enable	2GW
data inputs	2DA, 2DB

read enable	2GR
data outputs	2QA, 2QB

The following signal definitions define how the register file IC's are connected to form the multiport register file.

Section 1 of register file:

Input data (1DA, 1DB) - (XYINØ - XYIN11) The XCBUS (YCBUS) whose source is the LG FIFO.

Write address (1WØ, 1W1, 1W2) - WAØ, WA1, WA2 whose sources are the Character Control card (195223, slot 46, sheet 3) and the LG Input Control card (195218, slot 41, sheet 3), which are supplied via a tri-state bus. Note on the 195218 when the signal CHLOAD is high all three address bits are low, address Ø, which contains the XY input data for the new endpoint.

Write enable (1GW) - *INENABLE (C36) for both the X Adder Shifter, slot 39, and the Y Adder Shifter, slot 40, are generated on the 195218, LG Input Control card, slot 41, sheet 4. *INENABLE X for the X Adder Shifter and *INENABLE Y for the Y Adder Shifter.

Read address (1R0, 1R1, 1R2) - RAØ, RA1, RA2 from the ARITHCTLREG located on the Arithmetic Control card (195216, slot 38, sheet 1)

Read enable (1GR)	- Permanently enabled, tied low.
Output data (1QA, 1QB)	- (AØ - A11) The "A" inputs to the 74S181 ALU's.

Section 2 of the register file:

Input data (2DA, 2DB)	- (FØ - F11) The 12 bit output of the ALU.
Write/Read address (2W/RØ, 2W/R1, 2W/R2)	- RWBØ, RWB1, RWB2 whose source is the ARITHCTLREG on the Arithmetic Control Card (195216, slot 38, sheet1).
Write enable (2GW)	- *WRITE B whose source is the ARITHCTLREG on the 195216 slot 38.
Read enable (2GR)	- Permanently enabled; tied low.
Output data (2QA, 2QB)	- (BØ - B11) The "B" inputs to the 74S181 ALU's.

The arithmetic unit, ALU is also shown on sheet 1. The source of the function control bits, *SØ - *S3, is the ARITHCTLREG shown on sheet 1 of the Arithmetic Control, 195216. The source of the carry input to the ALU, *CARRYIN, is a flip flop on the Arithmetic Control card 195216, sheet 4. The ALU carry output is stored in the D flip flop shown at the left of sheet 3 of the 195217 logic. The output of these flip flops, *CARRYOUT, are sent back to the Arithmetic Control card 195216 where they are the

*XCARRYOUT and *YCARRYOUT inputs to the 2 to 1 selector of position 26, which is shown on sheet 4 of the 195216 logic.

The 195217-900 block diagram shows the interconnection of the eight word register file and the ALU described above. Section 1 of the register file receives its input data from the XCBUS or YCBUS. The input data to section 2 is the ALU, output, F. The A input to the ALU is the output of register file section 1 and the B input is the output of section 2.

The four control bit codes (*S0 - *S3) and carry input for the ALU and the operation it performs are:

*S3	*S2	*S1	*S0	CARRY IN	OPERATION
1	1	1	1	1	F ← A
Ø	1	1	Ø	1	F ← A + B
1	Ø	Ø	1	Ø	F ← A - B
1	1	Ø	Ø	Ø	F ← Ø

Note that the ALU only performs the functions of passing the A input, addition, subtraction and producing a zero output.

The contents of the two sections of the register file by address are shown in the MULTIPORT REGISTER ASSIGNMENT table on sheet 1 of the Arithmetic Control state diagram, 195216-950.

Sheet 2

Shows the maintenance register which allows selected signals from this card to be placed onto the XCBUS and YCBUS for use in hardware diagnostic programs.

Sheet 3 and Sheet 4

The logic for block normalization of ΔX and ΔY to form NΔX and NΔY is shown on these sheets. Block normalization arithmetically consists of left shifting the number until all but one zero has been shifted past the MSB. This normalization assumes that the numbers are positive; negative numbers are complimented before the shift. An example of a block normalized number is:

UNORMALIZED	000010111
NORMALLIZED	010111000

The normalization code is 3, the number of left shifts.

The logic for determining the number of left shifts is shown on sheet 3. The ΔX and ΔY calculated by the ALU are the F11 - FØ inputs to the 74S86 exclusive OR's which are used to compliment the F input if it is negative; if *BF12 is high. *BF12 is determined by the logic shown at the lower left of sheet 3. This logic determines if the output of the ALU is positive or

negative by examining the signs of the inputs, A11 and B11 and their relative magnitudes. Relative magnitude is obtained by comparing the carry outputs of the ALU, *CARRY. The ALU function bits *S1 and *SØ are sufficient to determine if the ALU is adding or subtracting; see the function table above.

The 74148, 8 to 3 line priority encoders in positions 32 and 33 and the 74S257 multiplexer determine the bit position of the left most one. These encoders are wired to look at the number from the upper bits downward. They output a code, A2 - AØ, which gives the input position of the first low input from the most significant bit downward. For example if the input F number has its upper four bits F11, F1Ø, F9 and F8 zero and bit F7 a one the upper encoder, in position 32, detects the fact that its inputs 7, 6, 5 and 4 are low and 3 is the first high input encountered. It outputs the A2 - AØ code Ø11; binary 3. Notice that this code nicely indicates that the first one encountered will have to be shifted three places left to occupy the MSB, i.e. one below the sign bit. Because at least one of its inputs is high the EØ output of 32 is high selecting the B inputs for the 2 to 1 multiplexer whose output is ENCODE (3 - Ø). If the inputs F11 through F7 are all low, 32.15, EØ is low and the multiplexer selects A, the outputs A2 - AØ of the lower, position 33, priority encoder.

The shift code, ENCODE (3 - Ø), for both ΔX and ΔY are output

to the 195215, Root Calculator where they are compared. The smaller of the two, XENCODE or YENCODE, is selected and placed on the ENCODE bus for both Arithmetic - Shifter cards. Hence the same shift code is used for normalization of both ΔX and ΔY. The control signal FINDSHIFT, sheet 3, lower right corner, controls the ENCODE compare and shift operation. The function of FINDSHIFT and the connection of the ENCODE bus to the 195215 card is described in detail in the 195215, Root Calculator card description of Section 4.5 above.

The selected ENCODE code, or value of N, is also used to denormalize N $\sqrt{(\Delta X)^2 + (\Delta Y)^2}$ on the 195215 and on the 195216, Multiplier Controller to specify if a vector is a long vector. The ΔX and ΔY normalizers are shown on sheet 4. They are implemented by using two levels of AMD, AM25S1Ø, four bit shifters which have tri-state outputs. The inputs are the twelve B bus or B inputs to the ALU lines; B(11 - Ø) and BSIGN. The shift input is the smaller of XENCODE and YENCODE which is now on the ENCODE bus as explained above. These shifters are wired to perform a left shift with the least bit replaced by the sign bit. Note carefully that the outputs NB(11 - Ø) shown at the bottom of the sheet are juxtaposed.

Sheet 5

The X and Y LENGTH REGISTERS and the X and Y POSITION registers

are shown on sheet 5.

The LENGTH REGISTERS store NΔX and NΔY. They have both a tri-state output, the Y's, which output to the 13 XYOUT or XY bus and totem pole outputs, the Q's whose 9 bits, NDXY(12 - 4) are sent to the squaring ROM's on the 195215 Root Calculator.

The X and Y Position Registers are also 25S18's. Only their tri-state outputs are used and they are connected directly to the 13 bit XYOUT or XY bus on this card.

The control signals applied to the position registers come from the MULTCTLREG on the 195216. Their function and name on the 195216 are:

a) *ENPOSOUT - (enable position out) the signal *XYPOSTOXYBUS, 195216 sheet 3.

b) *ENXYOUT - (enable XYOUT, i.e. NΔX, NΔY onto XYOUT bus) signals *DXTOXYBUS, *DYTOXYBUS, 195216 sheet 3.

c) *LDARITHOUT - (apply input clock to load both registers). The signal *MULTINPUT, 195216 sheet 3.

4.5 195212 X-Y ANALOG CARD

Functions Performed

a) Converts the twelve bit COS θ (X POSITION) and SIN θ (Y POSITION) inputs received from the X-Y Multiplier Card, 195214, to an analog voltage.

b) Integrates the analog voltage to generate the X and Y deflection inputs for the display.

c) Performs the extend integration that produces the X and Y CRT beam overshoot and undershoot required to insure that the "ends" of the line drawn on the CRT have the same brightness as the middle portion of the line.

References

a) 195212-900 - X-Y Analog Card Block Diagram

b) 195212-600 - X-Y Analog Card Logic Diagram

Sheet 1: X and Y - D/A Converters

The twelve bit COS θ, SIN θ or X POSITION and Y POSITION inputs received from the X-Y Multiplier Card, 195214 are loaded into the X Input Register, positions 13 and 14 and Y Input Register, positions 10 and 11 when *LOADOUTPUT goes low. This signal is generated on the 195213, sheet 1, and is ex-

plained in section 3.5 above. The outputs of the X and Y registers are applied to the twelve bit MDSL 1250 digital to analog converters. The MDSL 1250 is a high speed current output D/A converter with a maximum output current of 5.1 ma. The 200 OHM potentiometers, R1 and R2 provide output offset adjustment for the D/A converters.

Sheets 2 and 3: Buffers, Deflection Integrators and Extend Integrators

The simplified logic diagram of Figure 4.5-1 identifies the major functional entities of the circuitry shown on sheets 2 and 3 of the logic. It should be studied in conjunction with sheets 2 and 3 of the logic. In the following discussion the X circuitry of sheet 2 is described, the Y circuitry shown on sheet 3 is functionally identical.

Buffer Amplifier

The output current of the MDSL 1250 D/A converter, XDAC, is applied to the inverting input of the MSK-730 operational amplifier in position 22. The OP. AMP. has a 2.32K OHM feedback resistor and is thus connected in the standard inverting configuration. The maximum output voltage of the MSK-730 will thus be 5.1 ma X 2.32 KΩ = 11.8 volts as the maximum current output of the 1250 is 5.1 ma. Thus the analog voltage representing the magnitude of COS Θ and SIN Θ can be observed at the outputs of the MSK-730's which are connected to test points T7 (XDAC) and T3 (YDAC) which are located at the upper and lower rear of the card.

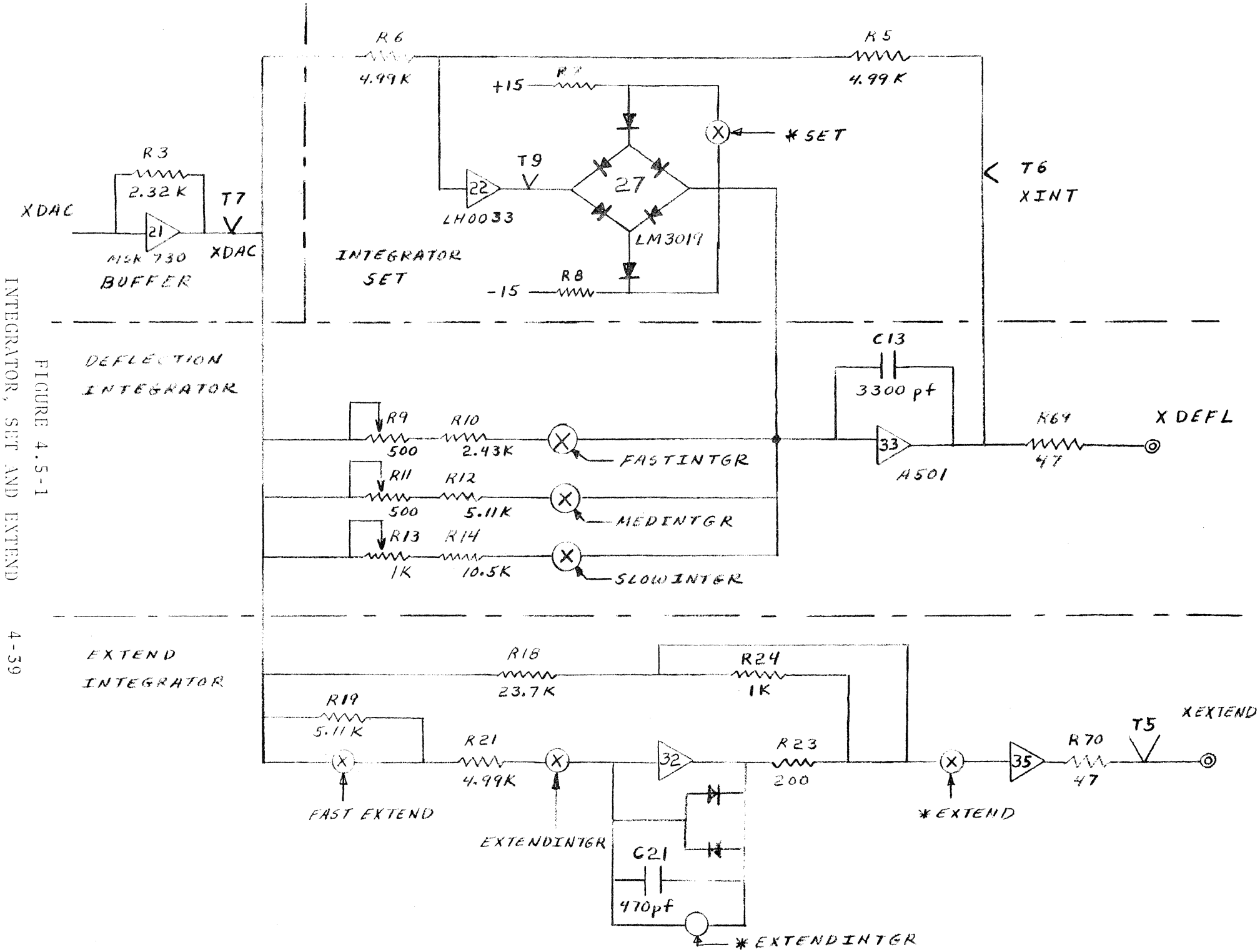


FIGURE 4.5-1 INTEGRATOR, SET AND EXTEND

Deflection Integrator

The deflection integrator consists of the A501 OP. AMP., position 33, the integrating capacitor C13, 3300 pf, and three input current paths. The input current paths each contain a potentiometer R9, R11 and R13, an input resistor R10, R12 and R14, a pair of back to back connected limiting diodes CR3-CR4, CR5-CR6 and CR7-CR8 and an SD5001 FET switch. The three input current paths provide three different ramp or deflection integration rates FASTINTGR, MEDINTGR and SLOWINTGR. Three paths are provided to allow this card to generate the deflection input voltages for both black and white and beam penetration-color CRT displays. The FASTINGR path is used with black and white displays. The MEDINTGR and SLOWINTGR paths are used when a beam penetration-color display is being driven. MEDINTGR is slected when a green vector is to be drawn and SLOWINTGR when a red or orange vector is to be drawn. The selection is done by the 5001 FET's. These FET's are off, open, when their gate input voltages, FASTINTGR, MEDINTGR or SLOWINTGR are zero volts and closed when their gate voltages are high, approximately 15 volts. Under proper operation only one of the FET's will be closed, have a high gate voltage applied.

The equation for output voltage of the deflection integrator, V_{XINT} is:

$$V_{XINT} = \frac{I_{IN}\, t}{C} \qquad (4.5\text{-}1)$$

As $$I_{IN} = \frac{V_{XDAC}}{R_{IN}} \qquad (4.5\text{-}2)$$

the ramp rate, volts per second is given by:

$$\frac{V_{XINT}}{t} = \frac{V_{XDAC}}{R_{IN}\,C} \qquad (4.5\text{-}3$$

Proper operation of the integrator can be checked by observing the slope of the output V_{XINT} with an oscilloscope for a known input voltage V_{XDAC}. The procedure consists of observing the input voltage, V_{XDAC}, at T7 or T3 and the output ramp V_{XINT} at T6 and or T2 with a test pattern such as the concentric boxes pattern. The range of the ramp rate volts per microsecond can be calculated with the aid of Table 4.5-1 which has been developed from equation 4.5-3 for the minimum and maximum values of R_{IN}.

INPUT PATH	SLOP OF V_{XINT} (VOLTS/MICROSECOND)
FASTINTGR (Black and White)	$.103\ V_{XDAC} \leq \frac{V_{XINT}}{t} \leq .125\ V_{XDAC}$
MEDINTGR (Green)	$.059\ V_{XDAC} \leq \frac{V_{XINT}}{t} \leq .054\ V_{XDAC}$
SLOWINTGR (Red)	$.029\ V_{XDAC} \leq \frac{V_{XINT}}{t} \leq .026\ V_{XDAC}$

TABLE 4.5-1
INTEGRATOR RATES

The back to back diodes connected to ground between the FET's and the input resistors limit the voltage swing to approximately ± 400 millivolts.

Connected to the drains of the three input FET's is a noise balancing network which consists of a 1.5-4 pf capacitor, C18 and C36, whose other end is connected to the signal BALANCE. This capacitor is adjusted to minimumize the effect of integrator switching noise.

Integrator Set Logic

The SET integrator logic is shown on Figure 4.5-1 and sheets 2 and 4 of the logic directly above the integrator logic. The function of this logic is to set or maintain the integrator output at its final desired level. When the integration time period is complete and the input, V_{XDAC} , is removed the integrator output will drift due to the leakage of various components. The SET logic is a feedback circuit which supplies current directly to the integrator OP. AMPS inverting input for the purpose of maintaining the output voltage constant.

The feedback path from integrator output to input consists of the two 4.99K resistors R5 and R6. The junction between the two resistors is connected to the input of the LH0033 voltage follower in position 22. The output of the voltage follower, which can be observed at test point T9, is input to a switch. This switch consists of the LM3019 diode array in position 27, the resistors R7 and R8 connected between plus and minus 15 volts and the SD211 FET switch Q_1. When the FET switch input *SET is high the FET is conducting and

the diode array is shorted. Thus the output of the voltage follower sees an open switch. When *SET is low the FET, Q_1, is an open switch. The balanced diode array looks like an open switch to the voltage follower and current is supplied to the inverting input of the integrator OP. AMP.

The Extend Integrator

In order to draw a line or vector of equal or uniform brightness on a CRT the beam must be deflected at a constant rate. A problem exists at the starting end and finish end of a vector. The CRT beam must be accelerated from a rest position, zero velocity, to the proper velocity and then decelerated at the finish end from the desired velocity to zero velocity. The acceleration must be done before the beam reaches the starting point and decelerated after it passes through the desired finish point in order to avoid vectors which have "brighter" end points. This process is accomplished by extending the beam position forward past the desired end point. Hence it also then starts before the desired start point of the next vector.

The extend integrator is shown at the bottom of sheets 2 and 3 of the logic and Figure 4.5-1. The MSK-730 in position 32 and 29 are the integrator OP. AMP. C21 and C39, 470 pf, are the integrator capacitors. The two 1N4148 back to back diodes across the integrating capacitor will limit the maximum voltage across the capacitor to a single diode drop. The input current path to the integrator consists of R19, 5.11K and R21, 4.99K and the SD5001 FET switch, EXTENDINTGR. The R19 input resistor has a TIS-74 switch, Q_2, across it. When the signal FASTEXTEND is high, 0 volts, the TIS-74 is on or closed and R19 is bypassed. This allows the input current to the OP. AMP. to be more than twice the value it has when Q_2 is open as the

input resistance on the switch open condition is 10.1 K and for the closed position it is 4.99K. The signal EXTENDINTGR operates the SD5001 in series with the OP. AMP input. When it is high the switch is on or closed and current is supplied to the integrator. The compliment signal, *EXTENDINTGR is applied to the SD5001 which is directly across the integrating capacitor C21. Thus when EXTENDINTGR goes low C21 is discharged and the integrator is reset to zero volts output. The *EXTEND signal when low opens the SD5001 at the input of the LH0033 voltage follower in position 35 and allows the XEXTEND output to be sent to the scope driver.

Figure 4.5-2 shows a simplified model of the extend integrator.

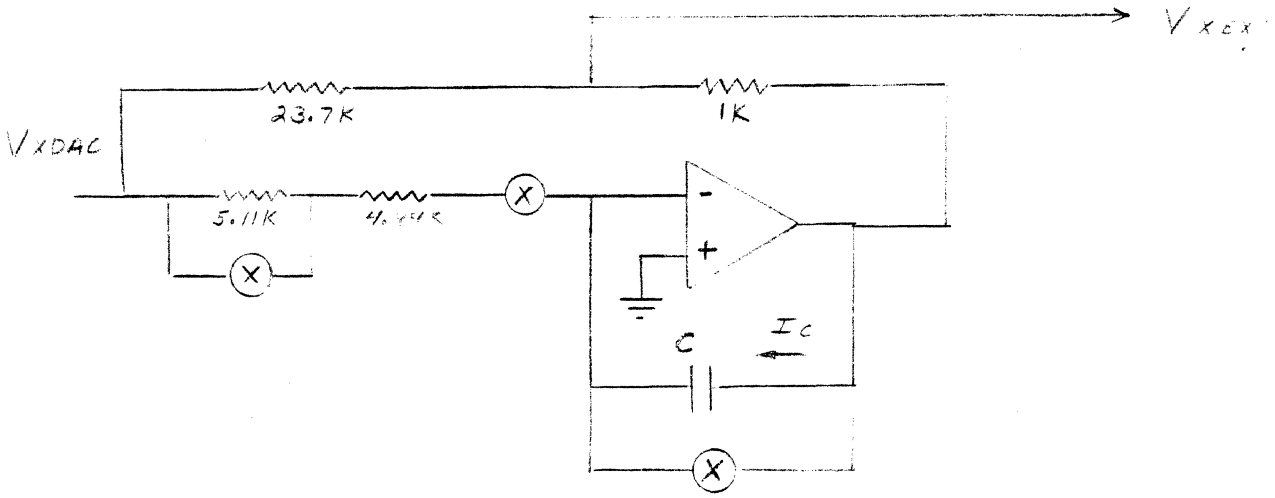


FIGURE 4.5-2
SIMPLIFIED EXTEND INTEGRATOR

Using this model the equation for the X EXTEND output is:

$$V_{XEXTEND} = V_{XDAC} \frac{1}{24.7} + \frac{23.7}{24.7} \frac{I_c}{C} t$$

The first term yields a voltage of approximately $\frac{1}{25} V_{XDAC}$. The second term indicates that a ramp will be generated when the EXTENDINTGR switch is closed, EXTENDINTGR goes high, and this ramp will be abruptly terminated when the *EXTENDINTGR switch is closed. ie C21 is shorted by the FET when EXTENDINTGR returns low.

Figure 4.5-3 shows the waveforms that will be observed at the test points XDAC (T7), XINT (T6) and XEXTEND (T5) for three sides of the concentric boxes pattern shown at the top of the figure. Below these waveforms is shown the waveform for the final X deflection, V_{XDEF}, which is the sum of V_{XINT} and $V_{XEXTEND}$. (Note that it is shown uninverted; when the addition is performed on the Scope Driver an inversion is performed). The waveform shown at the bottom is *SET and is included to indicate when the integrator set is performed.

Sheet 4: Generation of Switching Control Signals

The logic for generating the control signals for the SD5001, SD211 and TIS-74 FET switches is shown on this sheet. Table 4.5-2 lists the FET switch signals by name and shows the levels required to close (turn on) and open (turn off) the FET's.

The 75110 Dual Line Drivers are employed to generate the FET switching levels from standard TTL levels. The FASTEXTEND switching levels for the TIS-74 FET are generated by the DM-8800 shown at the lower left. The internal logic and the

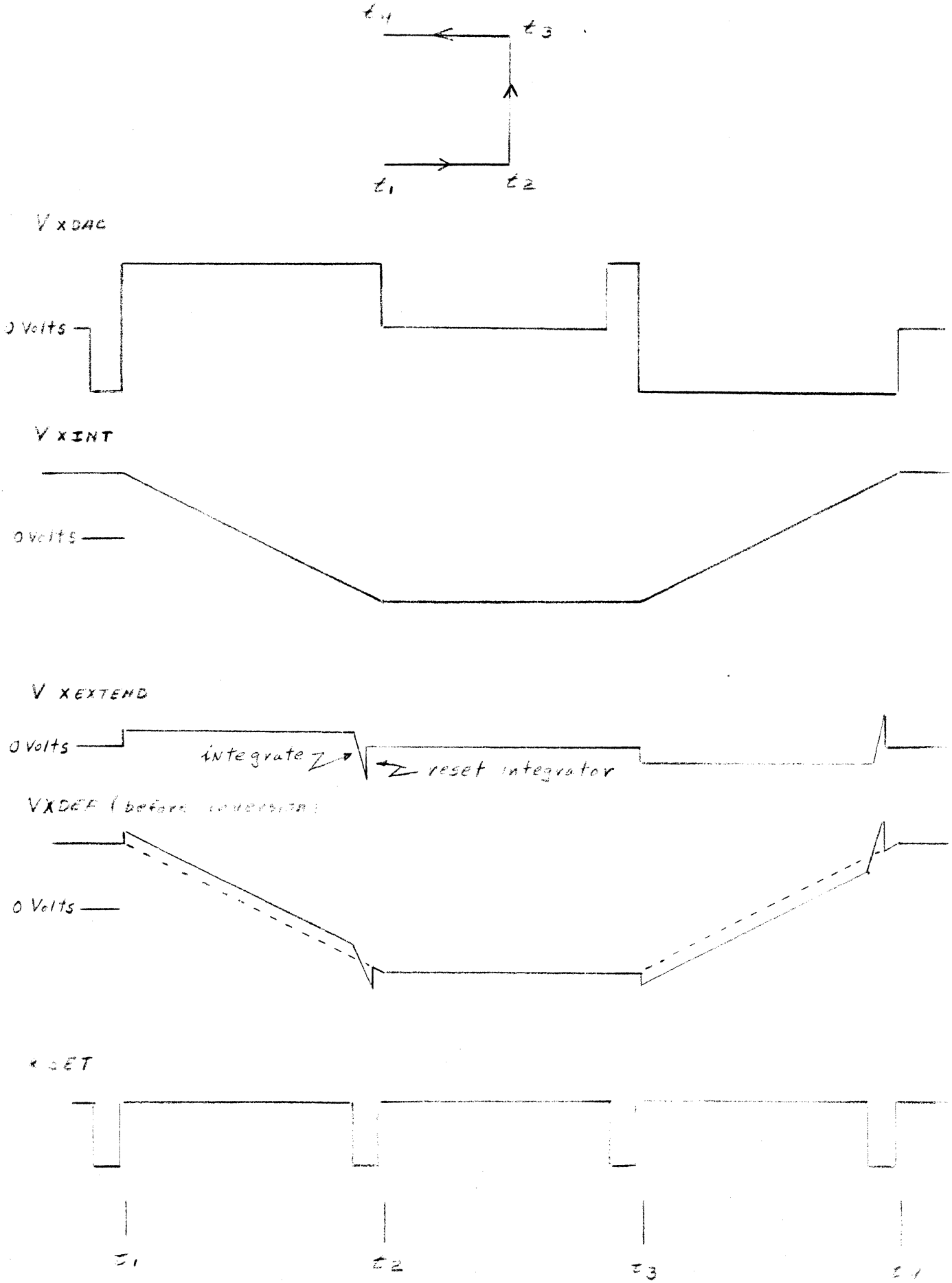


FIGURE 4.5-3

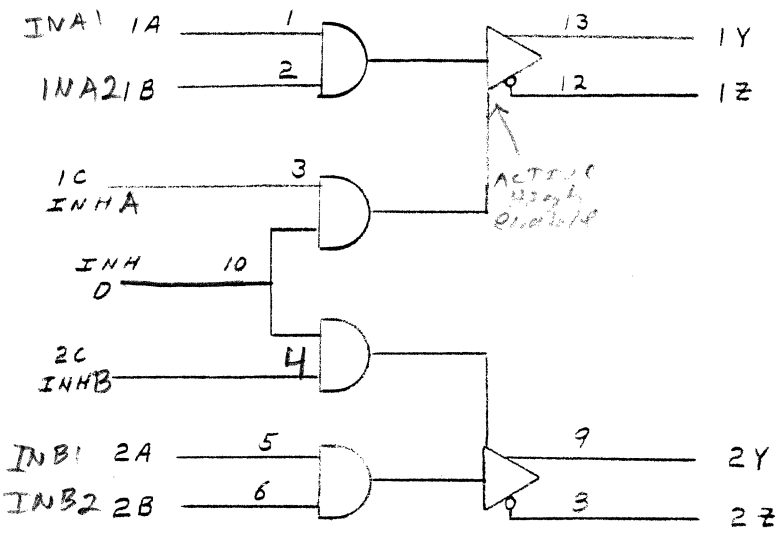
INTEGRATOR AND EXTEND WAVEFORMS

FET SWITCH	FET USED	SWITCH CONDITION AS FUNCTION OF INPUT LEVEL		
FASTINTGR MEDINTGR SLOWINTGR EXTENDINTGR *EXTENDINTGR *EXTEND	SD5001	High Low	10V 0V	Closed or On Open or Off
*SET	SD211	High Low	5V 0V	Closed or On Open or Off
FASTEXTEND	TIS-74	High Low	0V -6V	Closed or On Open or Off

TABLE 4.5-2
FET SWITCHING LEVELS

truth table for the 75110 is shown in Figure 4.5.4.

The 75110 in position 17 and the A half of the 75110 in position 20 generate the switching signals to select the deflection integrator speed. When the input INTEGRATE and one of the speed select inputs SLOW, MED or FAST are both high the respective output Y output SLOWINTGR, MEDINTGR or FASTINTGR will be high turning on or closing one of the three FET's which selects one of the three integrator input paths. Selection of any of the three paths causes the Z output and thus the signal BALANCE to the noise capacitors to go low. When the input FAST is high it causes the signal FASTEXTEND out of the DM8800 to be high which turns on the TIS-74 in the extend integrator input path causing the 5.11K resistors R19 and R43 to be bypassed.



LOGIC INPUTS		INHIBITOR INPUTS		OUTPUTS	
A_1	B_2	C_3	D_{10}	Y_{13}	Z_{12}
L or H	L or H	L	L or H	H	H
L or H	L or H	L or H	L	H	H
L	L or H	H	H	L	H
L or H	L	H	H	L	H
H	H	H	H	H	L

FIGURE 4.5-4
THE 75110 LINE DRIVER

The B section of the 75110 in position 20 is used to generate *SET which when low turns the SD211's, Q_1 and Q_3, off allowing the diode array switches, positions 27 and 23, to conduct thus setting the deflection integrator outputs.

The 75110 in position 25 is used to generate EXTENDINTGR and *EXTEND. Observe that the inversion of SLOW is applied to pin 25.1 and thus EXTENDINTGR is low when SLOW is high thus the extend integrator is not used when drawing lines at the slow or red speed.

The switching input signals to the 195212 are:

SET

This is SETANALOG which is generated on the 195213 Output Control card, sheet 1. Its generation is described on pages 3-60 and 3-61 above.

INTEGRATE

This is ANALOGINTEGRATE from the 195213. ANALOGINTEGRATE is generated on the 195213 by 'AND'ing INTEGRATE and INTEGRATE delayed by two 74S04 gate delays. (See sheet 5 of the 195213). INTEGRATE is generated on the 195215 sheet 4. It is high for the desired period of the integration and is returned low when the integration time counter overflows.

FAST

Is generated on the 195213, sheet 4, and is the output of the PROM A23 in position 20.

Observe that two of the address bits to this PROM are INTEN 1 and INTEN 0 which are the bits XCBUS 1 and XCBUS 0 after they are buffered in the flip flop in position 24 on sheet 4 of the 195216. These XC bits are the INTENSITY code bits of word 1 of the Line Generator STATUS word.

SLOW

Is generated in the same manner as FAST. It corresponds to the INTENSITY code, 11, of the Line Generator Status word.

MED

Is generated on sheet 5 of the 195213 and will be high if both FAST and SLOW are low.

*EXTENDIN

Is the signal *EXTEND generated on the 195213, sheet 2. Its generation is described on pages 3-58 through 3-60 above.

EXTENDINTGR

Is generated on the 195213, sheet 2 and is also described on pages 3-58 through 3-60.

4.6 195219 LINE GENERATOR FIFO PORT

Function Performed

The Line Generator FIFO accepts 16 bit input words from the PSBUS and assembles them into the 32 bit Line Generator input commands; MOVE or DRAW, STATUS and CHARACTER COMMAND. These words are described on pages 2-10 thru 2-13 above. The FIFO holds up to four 32 bit input command words.

This card decodes the Line Generator address, keeps track of the condition of the FIFO and generates the "flag" that tells the Line Generator Input Control on the 195218 that the FIFO has data input for the Line Generator.

References

195219-600 - L.G. FIFO PORT CARD

Sheets 3 and 4: The FIFO

The structure of the FIFO is shown in the block diagram of figure 4.6-1. It consists of the Left and the Right FIFO. The inputs are received as two sixteen bit words, Word 1 and Word 2 of the input command, and stored as a single 32 bit command in the FIFO. Observe that the Left FIFO will contain Word 1 and output to the XCBUS while the Right FIFO contains Word 2 and outputs to the YCBUS. The composite FIFO is 4 words by 32 bits. The maintenance multiplexer allows the XCBUS and YCBUS to be read back to the PSBUS for diagnostic purposes.

The Left and Right FIFO's are written into successively, the

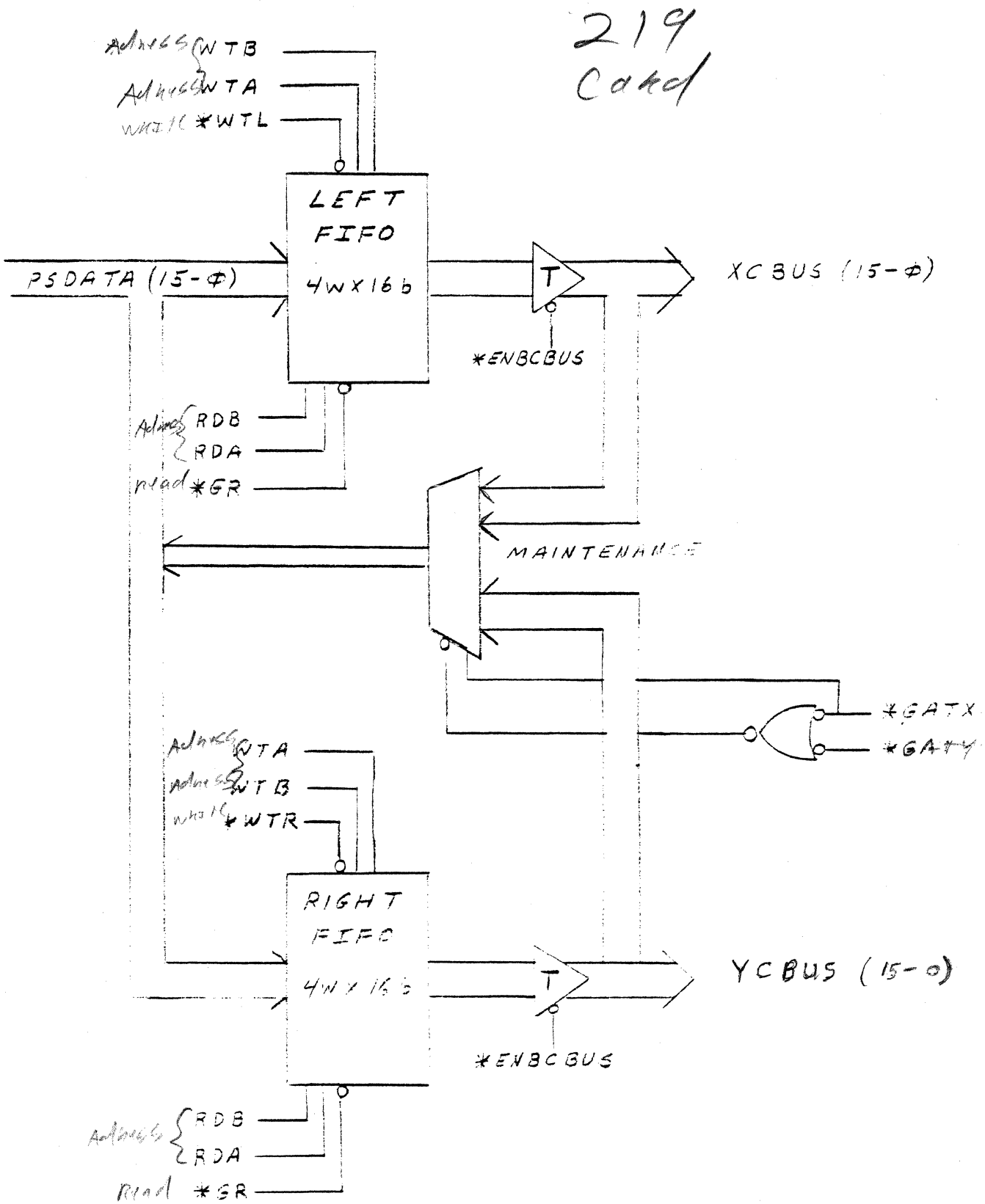


Figure 4.6-1
Line Generator FIFO Block Diagram

write signals being *WTL and *WTR. WTB and WTA are the write address inputs. Both FIFO's are read simultaneously onto the XCBUS and YCBUS. The read signal is *GR and the read address inputs are RDB and RDA. These signals are generated on sheet 1 and are described below. *ENBCBUS, when low, enables the FIFO onput onto the XCBUS and YCBUS. It is generated on the LG Input Control 195218 and is low except when the Character Generator uses the XCBUS and YCBUS or the Line Generator is in the maintenance mode.

Sheet 2: FIFO or Line Generator Address Decode

The Line Generator address detection logic is shown at the left of sheet 2. The PSADD lines are compared to the address set into the switch in position 11 by the DM8136 comparators in positions 21 and 26. The signals PSWRT, indicating that the PSBUS cycle is a write cycle, and PSDEV, indicating that the PSBUS is addressing the SCB, must also be true. The output of the flip flop 51.14 will be low if the addressing is valid. The output of 70.3, LDFIFO, will be high when the addressing is valid or when *FIFOLD from the refresh controller is low. LDFIFO is stored in the flip flop 51.11. Observe that *PSBSY(A) 70.3 goes low when the valid address is detected and remains low until one clock period after the address becomes invalid and thus is low for at least two clock periods. Observe that *PSBSY(A) will also be low if FIFULL is high. FIFULL indicates that the Line Generator FIFO is full, contains four 32 bit words, and cannot accept more input data from the PSBUS.

The logic shown at the bottom of sheet 2, positions 20, 40, 41 and 50 is used in the maintenance mode. The outputs of

the 74S175 flip flop, position 50, have the following functions:

*GATXX and *GATYY -

Respectively select and gate XCBUSS and YCBUSS onto the PSBUS.

LDLGST -

Sent to the 195218 Input Control card to load maintenance control prom address register, called POKEMNT on the 195218, see sheet 2.

GATLGST -

Sent to the 195218 to gate in maintenance inputs from PSBUS. Called READMNT on the 195218, see sheet 2.

Sheet 1: FIFO Control Logic

The FIFO control logic consists of:

a) The left/right FIFO indicator flip flop, *LEFT, in position 66.

b) The Write address counter in position 73 whose outputs are WTB and WTA.

c) The Read address counter in position 72 whose outputs are RDB and RDA.

d) The FIFO Status PROM (A70) in position 71.

e) The data input flag, flip flop *LGRDY, which indicates that the FIFO has data to output to the 195217 X-Y Adder-Shifter cards.

The state of the logic before a Line Generator input is to be written into the FIFO, address not yet decoded, finds both *LEFT, position 66, and LDFIFO low. Thus, the Write Address counter enable IWTCNT and the FIFO write enables *WTR and *WTL are both high. On the next clock after the LDFIFO flip flop, sheet 2, position 51, is clocked high *WTL goes low enabling a write into the Right FIFO. *LEFT is toggled, LDFIFO is applied to both J and K, and on the next clock *WTR goes low enabling a write into the LEFT FIFO. IWCNT is also high and the write counter is counted up.

The function of the FIFO Status PROM of position 71 is to keep track of the status of the FIFO. Essentially, three conditions exist for the FIFO:

a) It is empty. It can receive data from the PSBUS, be written into, but has no data for the Line Generator and cannot be read out.

b) It is full. It cannot receive data from the PSBUS, be written into, but it can be read out to the Line Generator.

c) It is not empty and it is not full. Thus, it can be both written into and read out.

The outputs of the FIFO Status PROM indicate the state of the FIFO. The outputs have the following meaning:

71.12 Empty - will be high when the FIFO is empty.

71.11 Not Empty - will be high when the FIFO is not empty.

71.10 FIFULL - high when the FIFO is full.

71.9 *FIFBSY - low when the FIFO is being read or written.

The contents of the Status PROM is shown on sheet 5 of the logic.

The FIFO Status PROM is addressed by the four bits of the Read counter, the lower three bits of the Write counter and the *LEFT flip flop. The counters are operated as ring counters and are only reset, cleared, by the Line Generator reset *RST. The relative counts of the Write and Read counters indicate the condition of the FIFO. As shown, the first two bits of the counters are RDB, RDA and WTB, WTA. For purposes of this discussion, we name the third bits RDC and WTC. The three possible conditions of the FIFO can now be expressed in terms of the counter bits.

a) FIFO Full.

$$[RDB,RDA] = [WTB,WTA]$$

and

$$RDC \neq WTC$$

The FIFO full condition exists when the lower two bits of the counters are equal and the third bits are not. This is equivalent to stating that four more writes than reads have occurred as the Read Counter is equal to the Write Counter plus or minus four counts. Observe from the PROM table of sheet 5 that output 03, FIFULL, is high for address inputs Ø4, 15, 26, 37, 4Ø etc. and 4Ø, 51, 62 etc.

b) FIFO Empty.

$$[RDC,RDB,RDA] = [WTC,WTB,WTA]$$

The FIFO is empty if the same number of reads as writes has been made. One other condition is accounted for by address bit A6 of the PROM which is connected to the QA output of the Read counter. This bit will be high when the FIFO has been read out, data on the output buss, but the Line Generator has not yet taken the data. The 02 output bit of the PROM, 71.11, will be high when the FIFO is not empty. The least bit of the PROM, 71.12, is the empty bit and is high when the FIFO is empty.

The 04 output of the PROM, 71.9, is *FIFBSY, or FIFO busy. It will be low when the FIFO is full, FIFULL high. It is also low for the clock period following the Writing into of the Right FIFO and insures that the next word will not be immediately sent over the PSBUS. This condition exists when:

[RDC,RDB,RDA] +1 = [WTC,WTB,WTA]

The flip flop in position 66 whose $\overline{Q}$ output is *LGRDY is the flag to the Input Control that indicates that the FIFO has an input for the Line Generator. This flip flop is J'ed by the not empty bit of the Status Prom, 71.11, and K'ed by the acknowledge back from the Input Control, *LGSTR, which goes low after the FIFO output has been accepted by the Line Generator.

8.0 Alignment and Tuning of the Line Generator

The following tuning procedure will result in a properly tuned Line Generator and Display provided all circuitry is functioning properly. Two L.G. cards are involved in the adjustments, X-Y ANALOG (195212) and SCOPE DRIVER (195211). The location of the relevant components are shown in the card layouts of Figure 8.20 and 8.21 the last pages of this section.

The following Test Equipment is required to perform the procedures:

Digital Volt Meter

Oscilloscope

Tuning tool for small capacitors and Potentiometers

Extender Cards - 2 each

Computer Test Routines

6 Ft. Coax, with T and 75 OHM Terminator

8.1 Start Up Procedure

a. Disable the extend logic by setting the printed circuit switches 1 and 2, located near position 24 on the Output Control card (195213-100), slot 35, as follows:

SW1 - open
SW2 - closed

b. Place XY Analog card, 195212-100, slot 33 on an extender.

c. Place a Scope Driver card, 195211-100, slot 1, 2, or 3, on an extender and connect a Display.

d. Turn on power and measure the voltages on the XY Analog card, back panel slot 33.

PIN 76 - +15v ± 0.03v (±0.2%)
PIN 73 - -15v ± 0.03v (±0.2%)
PIN 8 - 5v ± 0.05v (±1%)

Adjust Power Supplies if not within required voltage limits.

e. Load the Hardware Diagnostic Programs in the Picture Controller.

f. Turn the Display on with the INTENSITY and BRIGHTNESS controls set to min. (CCW). Increase INTENSITY and BRIGHTNESS only when viewing is required. Turn CONTRAST to OFF (full CCW). Allow 1/2 hour warmup.

8.2 X-Y ANALOG CARD (195212) Slot 33

8.2.1 Zeroing of the XY Integrator Output Offset Voltage

a. Use QSD020 to generate the 16 concentric circle pattern drawn at the slow (RED) speed. Figure 8.1.

QSD020 - Phase 1 - to select RED
- Phase 13 - to obtain circle pattern

b. Temporarily ground the X clamp diodes (at the point connecting CR7 and CR8) that limit input voltage swing to the O/OR/RED X-Integrator switch. See

Figure 8.20 pointer A. The concentric circle pattern will change and appear as shown in Figure 8.2.

c. If the short vectors are not vertical then adjust the X-Integrator operational amplifier offset potentiometer, R15, see pointer A Figure 8.20, until the short vectors are vertical.

d. Remove the ground between the clamp diodes CR7 and CR8 and temporarily ground the Y clamp diodes (at the point connecting CR20 and CR21) that limits the input voltage swing to the 0/OR/RED Y Integrator switch. See Figure 8.20. The concentric circle pattern will now appear as shown in Figure 8.3.

e. If the short vectors are not horizontal then adjust the Y-Integrator operational amplifier offset potentiometer, R39 Figure 8.20 until the short vectors are horizontal.

f. Remove the clamp diode ground.

8.2.2 X and Y DAC Output Offset Voltage and Integrator Input Current Adjust

Use QSD020 phase 12 to generate the 63 concentric box pattern. Phase 1 will have to be used to successively choose RED, GREEN and BLACK and WHITE speeds in the following procedure.

When DAC offset and YX integration lengths are properly

adjusted, the pattern will appear as shown in Figure 8.8. Due to drift the pattern may exibit a combination of mal-adjustments. The effects of the various trim potentiometers are distinctive making adjustment simple provided all steps a through c are followed in order as directed.

X and Y DAC offset

a. RED speed, SLOW INTEGRATOR. Turn the X DAC offset potentiometer, R1, Figure 8.20 pointer B, until all vertical lines, Figure 8.4, become parallel. Ignore other peculiarities.

Turn the Y DAC offset potentiometer, R2, Figure 8.20, pointer B, until all horizontal lines, Figure 8.5, become parallel. Ignore other peculiarities.

Slow (RED) Speed Integrator Input Current

Turn the O/OR/RED X-length potentiometer, ~~R14~~ R13, Figure 8.20, pointer B, until the horizontal lines just meet the vertical lines, Figure 8.6, (or the imaginary extension of vertical lines). If the upper and lower line sets do not match, balance the errors.

Turn O/OR/RED Y-length potentiometer, R37, Figure 8.20, pointer B, until the vertical lines just meet the horizontal lines, Figure 8.7. If the left and right vertical line sets do not match, balance the errors.

Repeat the above steps until the correctly adjusted pattern, Figure 8.8, is obtained.

Medium (GREEN) Speed Integrator Input Current

b. Using phase 1 select the color GREEN, using phase 12 again obtain the concentric box pattern.

Turn the G/Y X-length potentiometer R11 until the horizontal lines meet the vertical lines (or the imaginary extension of the vertical lines) as indicated in Figure 8.6.

Turn the G/Y Y-length potentiometer R35 until the vertical lines meet the horizontal lines, as indicated in Figure 8.7.

Repeat the above steps until the correctly adjusted pattern, Figure 8.8 is obtained. It should not be necessary to have to readjust either of the DAC offset potentiometers.

Fast (BLACK and WHITE) Speed Integrator Input Current

c. Using phase 1 select the color BLACK, using phase 2 again obtain the concentric box pattern. The pattern should appear dimmer than it did in the above as the beam is now moving at its fastest rates.

Turn the B/W X-length potentiometer R9 until the horizontal lines just meet the vertical lines (or the imaginary extension of the vertical lines) as indicated in Figure 8.6.

Turn the B/W Y-length potentiometer R33 until the vertical lines just meet the horizontal lines as indicated in Figure 8.7.

Repeat the above steps until the correctly adjusted pattern, Figure 8.8 is obtained. It should not be necessary to readjust the DAC offset potentiometer. Recheck all concentric square patterns (B/W, G, Y, O, or RED). If all do not appear like Figure 8.8, then repeat this tuning procedure form a through c.

8.3 DISPLAY Phase Adjustment

Use QSD020 phase 17 to generate the large X pattern. Two crossed sets of parallel lines drawn in opposite direction will appear as shown in Figure 8.9. If the lines do not overlay, tune the phase adjustment of the Display. The phase adjustment potentiometers are found on the X and Y Deflection Preamplifiers, R13. This is the only potentiometer on these cards.

After proper phase adjustment recheck the adjustments of X and Y DAC offset and integrator input current by observing the concentric squares pattern for RED, GREEN and BLACK and WHITE colors.

8.4 Further Adjustments on the X-Y ANALOG CARD - Compensation For Extend Integrator and X-Y Integrator Switching Noise

a. Turn the display BRIGHTNESS and INTENSITY down. Type control C's to exit the diagnostic.

b. Turn the Picture System off.

c. Remove the Output Control Card (195213) and enable the extend logic by resetting the printed circuit switches 1 and 2 as follows:

SW1 - closed
SW2 - open

8.4.1 Null Extend Integrator Switching Noise

Use QSD026, phase 14, to display the character set.

Examine the small characters to determine if the short vectors are straight. Crooked lines are indicative of Extend Integrator switching noise. See Figures 8.10 and 8.11. Capacitors C4 and C5 couple the Extend Integrator (MSK 730 OP AMP) modes with the inverse of the gate switching waveform and thus act to cancel the noise.

Adjust the X Extend Integrator noise coupling capacitor C4, pointer C, Figure 8.20, to obtain straight vertical vectors. Adjust the Y Extend Integrator noise coupling capacitor C5, pointer C, Figure 8.20, to obtain straight horizontal vectors.

8.4.2 Null X-Y Integraotr Switching Noise

Use QSD026, phase 14, to display all 95 characters in all eight signs. Adjust the X-Integrator balance capacitor,

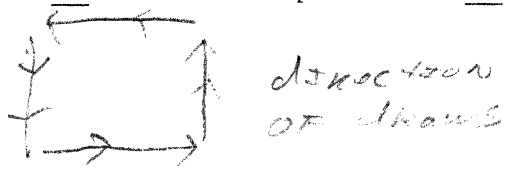
C18, and the Y-Integrator balance capacitor C36 for the best appearing 10-point characters. See Figure 8.20, pointer D.

8.5 SCOPE DRIVER (195211) END POINT MATCH

Before attempting to end point match the reader should be sure he understands the operation of the unblank delay line, the delay line tap switches and the "OR" logic on the 195211 card. This circuitry is fully described in the section; Display Unblank (sheet 2) of the 195211 card description found in section 4.1 of this manual.

For a display connected to the "A" outputs S2 is position 35 and S1 is position 44. For a display connected to the "B" outputs S2 is position 36 and S1 is position 45.

a) Set Up



Use QSD020 phase 12 to generate the concentric box pattern at the fast integrator speed. The objective of these adjustments is to obtain a correctly endpoint-matched box pattern as shown in Figure 8.12.

b) Set the Unblank Pulse Delay Time

Set the tap switches as follows:
S2 [position 35 (36)] - close switch position 10 only S1 [position 44 (45)] - all switches open.

The pattern should appear as shown in Figure 8.13a, NOT ENOUGH DELAY, which shows that the vectors "start too soon". Notice that the vectors of the concentric boxes pattern are drawn in the counter clockwise direction. Thus in Figure 8.13a the end points 1, 2, 3 and 4 are the start ends and a, b, c and d are the finish ends. Close position 9 and open position 10 of S2 noticing that the start ends of the vectors are further delayed. Try closing successively positions 8, 7, 6.... until the start ends of the vectors are correctly delayed. The correct pulse delay is shown in Figure 8.13b, CORRECT DELAY, where the start ends of the vectors are properly aligned with the finish end or the finish end extended of the preceding vectors. Figure 8.13c shows the case where the pulse delay is too long, TOO MUCH DELAY, note how the vectors start too late. The TOO MUCH DELAY situation occurs when the correct position of S2 has been passed.

c) Set the Unblank Pulse Width

This adjustment selects an Unblank Pulse Width such that the finish end of the vectors are delayed the proper amount to close the gaps between the finish end of a vector and the start end of the next vector drawn. Figure 8.13b shows these gaps.

Note the position of S2 that was closed to obtain the correct pulse delay in the above adjustment of the start ends. Close a position of S1 one position less. eg. if S2 had position 5 closed start with position 4 of S1 closed. Notice the effect on the finish ends of the vectors as successively lower numbered position of S1 are closed. Figure

8.14a, NOT WIDE ENOUGH, shows the condition where the pulse width is too narrow, Figure 8.14b, TOO WIDE A PULSE, shows the condition where the pulse is too wide, the finish ends extend too far. The correct delay and end point match is shown in Figure 8.12. The specifications for PS-2 calls for an endpoint match of within .020 inches. Figure 8.15b, shows the definition of the cases for both overshoot and undershoot both of which should be .020 inches or less.

d) Verify End Point Match

Verify the end point match by using QSD026 phase 14 to display the full character set. Examine the end point match on the small and medium size characters. It may be desirable to vary the settings of S2 and S1 one position to optimize the end point match for the characters. Generally this will cause a noticeable effect on the concentric boxes pattern. Be sure to recheck the boxes by running QSD020 phase 12 if you vary either S2 or S1 to optimize the character end point match.

Both the boxes and the characters must possess an end point match within the specified .020 inches. However it is possible on most machines to achieve a better end point match for either short vectors, the characters, or long vectors, the boxes and becomes a matter of choice as to which you choose.

e) Inability to Achieve .020 inch End Point Match

If the specified end point match can not be achieved using the above procedure one or both

of two conditions exist:

1. The offset and integrator input current adjustments of the X-Y Analog Card, 195212, are incorrect. Repeat the adjustments of sections 8.2 and 8.4.

2. A component on the X-Y Analog card is out of tolerance. In most cases this will be evidenced by inability to achieve the desired results in the adjustments of sections 8.2 and 8.4. If this condition exists exhaustive troubleshooting of the X-Y Analog card is required.

8.6 SCOPE DRIVER (195211) - NO CONTRAST

8.6.1 Display Controls and Scope Driver Potentiometers

Before performing the brightness and contrast adjustments it is important to review the function of Display Controls and the potentiometers on the Scope Driver and the Display Video Amplifier. You should be sure you understand the following:

DISPLAY INTENSITY CONTROL - This potentiometer controls the cathode to grid voltage bias on the CRT. See the Xytron Video Amplifier schematic, R29.

DISPLAY BRIGHTNESS CONTROL - This potentiometer controls the video level presented to the display Video or Z Axis Input. It is wired as shown in Figure 8.16.

These two controls are accessable to the user and you should instruct him in their use:

DISPLAY INTENSITY CONTROL - Is used as a "safety switch" to quickly cut off the image. The Display should be operated with this control set to its full CW position. Any other setting will cause short and long vectors of equal brightness to appear unequally bright. You can in fact demonstrate this by observing the effect of varying the INTENSITY CONTROL on the concentric circles pattern, QSD020 phase 13.

DISPLAY BRIGHTNESS CONTROL - This is the control the user should employ to set the overall brightness of the Display at a desired level.

The remaining three controls are on the Display Video Amplifier card and on the 195211 Scope Driver, and are therefore; not accessable to the user.

DISPLAY THRESHOLD ADJUST - This is the most critical adjustment, it is on the Display Video Amplifier. The setting of this potentiometer is critical to providing smooth brightness control and equal vector brightness. It determines the "gain threshold" for the Display Video Amplifier. (Ref: Xytron Video Amplifier Schematic R10).

SCOPE DRIVER CONTRAST ADJUST - This potentiometer controls the gain of the video amplifier on the 195211 Scope Driver by

determing the amount of current input to the MSK 730. (Ref: 195211 R69 and/or R76). Its adjustment is described in section 8.7 below.

SCOPE DRIVER MAX BRIGHTNESS - This potentiometer sets the output offset voltage of the video amplifier on the 195211 and as mentioned above has been deleted on new versions of the card. Its adjustment is discussed in the attached modified intensity, brightness and contrast procedure.

8.6.2 NO CONTRAST - EQUAL VECTOR BRIGHTNESS ADJUSTMENT

These adjustments should be made with the room lighting level at that which the customer desires to operate the system.

a. Set the Display CONTRAST control to the "no Contrast" position (fully CCW). Set both INTENSITY and BRIGHTNESS to maximum (both controls fully CW).

b. Using QSD020, phase 13, display the concentric circles, Figure 8.17, pattern with BLACK and WHITE color; FAST INTEGRATOR.

c. Adjust the Scope Driver Contrast Adjust, R69 and/or R76, Figure 8.21, fully CW for maximum gain. The concentric circles pattern should appear at maximum brightness.

d. Remove the Video coaxial cable from the display and turn the BRIGHTNESS control to minimum brightness (full CCW). Leave the INTENSITY control set to full intensity (full CW).

e. Adjust the THRESHOLD potentiometer on the Display Video Amplifier card (the only control on this card) until the concentric circle pattern is just visible. It should be possible to completely cut the beam off (no visible image) and to obtain a fairly bright image by taking the THRESHOLD potentiometer through its entire range. If this is not possible there is probably a failure on the Video Amplifier card.

f. Reconnect the Video coaxial cable to the Display. The pattern should now appear at maximum brightness.

g. Turn the INTENSITY control on the Display and observe how the small (1/2 and 1/4 speed) lines, Figure 8.17, reduce in intensity more rapidly than the longer (full speed) lines.

Return the INTENSITY control to maximum (CW). Vary the BRIGHTNESS. When the THRESHOLD is properly adjusted, all lines should have equal intensity at all brightness levels. (Except that operation becomes non-linear and thus unuseable when the BRIGHTNESS control is below 1/4 of its maximum CW position).

The procedure steps "d" through "g", may have to be repeated several times to obtain equal brightness of all vectors with the BRIGHTNESS control due to variations in the gain of different CRT's.

As a final check run QSD026, phase 14, and observe that for the full character set all vectors are of equal brightness as the BRIGHTNESS control is varied.

8.7 SCOPE DRIVER (195211) CONTRAST ADJUSTMENTS

8.7.1 Video Delay

This adjustment is only required when the Line Generator contains the Z Depth Que Card, 195231, which provides a contrast capability for the system.

a. Using QSD020, phase 12, display the concentric squares pattern with BLACK and WHITE color, FAST INTEGRATOR.

b. Set the CONTRAST switch on the Display to the first contrast position (one step CW). The set of nested squares, Figure 8.18 will appear with intensity of each line smoothly varying. The Scope Driver Video delay line tap switches (position 55 for "A" and position 58 for "B") should be ajusted until lines appear symmetrical around a 45° diagonal (imaginary) line.

c. Check the remaining four CONTRAST switch settings to insure that they are operable.

d. If no change is observable on the display or if the contrast does not appear to be operating correctly use an oscilloscope to observe the CRT cathode waveform, Figure 8.19. (The cathode is connected to the Video Amplifier card by the red wire between the CRT socket and the amplifier card). The observed voltage waveform should have approximately the voltage values indicated in

Figure 8.19. Note that the values are different for different CRT's, Thomas and Raytheon, and in-fact can be expected to vary from CRT to CRT. The magnitude of the video voltage swing is changed by adjusting the Scope Driver Contrast Adjustment R69 and/or R76 on the 195211. Note that this is the voltage gain adjustment for video amplifier (MSK 730). Effectively this gain adjustment provides a variable video output voltage swing in order to "match" the Scope Driver to different CRT's.

The -100 and -101 versions of the 195211 and -102 and -103 below ECO level A4 contain an offset potentiometer, labelled MAX BRIGHT ADJUST on the logic and VIDEO OFFSET on the card. This potentiometer should be adjusted to obtain the lower value, the FULL INTENSITY voltage, of 7 or 30 volts shown in Figure 8.19. This offset adjustment thus controls the maximum brightness of the image.

8.7.2 Video DAC Glitch Attenuation

a. Using QSD020 display the set of five varying intensity raster bars, phase 5.

b. Without compensation, a vertical glitch may appear about 1/7 of the distance from the maximum intensity end of a given raster set, and a smaller vertical glitch may appear about 1/14 and 2/7 the distance from the maximum intensity end. Variation of two capacitors near the appropriate DAC located on the Scope Driver card will

nearly eliminate these glitches. Without compensation, glitches are bright for a decreasing Z ramp and dark for an increasing Z ramp.

For adjustment of the "A" Display output, adjust capacitor C56, the capacitor between the MSB digital input to the DAC (41.5) and the node (41.2) until the larger glitch disappears. Adjust capacitor C57 between the second MSB digital input to the DAC (41.6) and the node (41.2) until the smaller glitch disappears. Repeat adjustment of both capacitors until glitches are suppressed.

c. For adjustment of the "B" Display output, the MSB capacitor, C58, is between pins 42.5 and 42.2, while the second MSB capacitor C59, is between pins 42.6 and 42.2. Eliminate the larger glitch by adjusting C58, and the smaller glitch by adjusting C59.

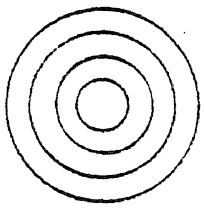


FIG. 8.1
Concentric Circles

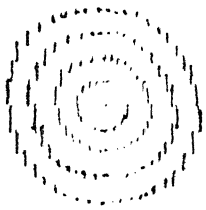


FIG. 8.2
X Integrator Output Offset Adjustment

RED, fast integrator speed
Ground input between
CR7 & CR8
Adjust R15
for verticle short vectors

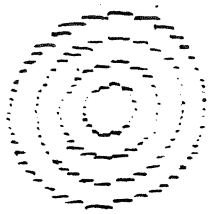
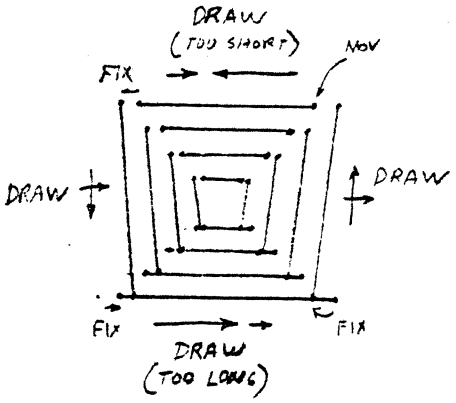


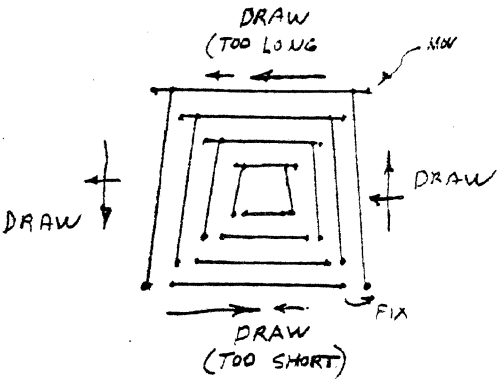
FIG. 8.3
Y Integrator Output Offset Adjustment

RED, fast integrator speed
Ground input between
CR16 & CR17
Adjust R39
for horizontal short vectors



X DAC
Offset Positive
Adjost R1 CCW

Finnish position is to the RIGHT of the correct position if line start point is corrected.

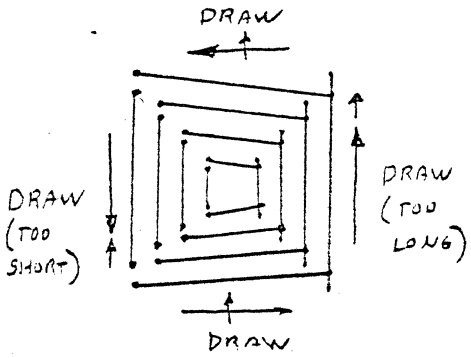


X DAC
Offset Negative
Adjust R1 CW

Finnish position is to the LEFT of the correct position if line start point is corrected

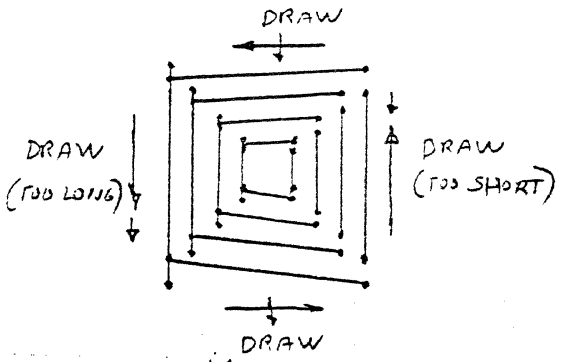
FIG. 8.4

X DAC
Offset
Adjustments



Y DAC
Offset Positive
Adjust R2 CCW

Finish position is ABOVE the correct position if line start point is corrected



Y DAC
Offset Negative
Adjust R2 CW

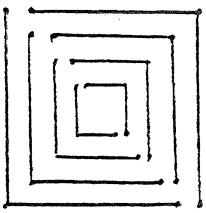
Finnish position is BELOW the correct position if line start point is corrected

FIG. 8.5

Y DAC
Offset
Adjustment

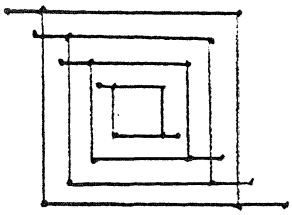
FIG. 8.6

X LENGTH ADJUST
(x integration rate)



X LENGTH TOO SHORT

B/W: TURN R 9 CW
G/Y: TURN R 11 CW
O/OR/R: TURN R 13 CW

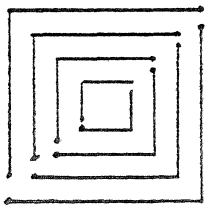


X LENGTH TOO LONG

B/W: TURN R 9 CCW → 4th pot from top
G/Y: TURN R 11 CCW → 3rd pot from top
O/OR/R: TURN R 13 CCW → 2nd pot from top

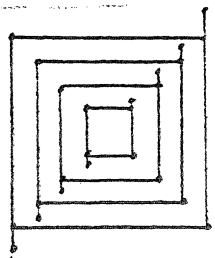
FIG. 8.7

Y LENGTH ADJUST
(Y integration rate)



Y LENGTH TOO SHORT

B/W: TURN R 33 CW
G/Y: TURN R 35 CW
O/OR/R: TURN R 37 CW



Y LENGTH TOO LONG

B/W: TURN R 33 CCW → 4th pot from bottom
G/Y; TURN R 35 CCW → 3rd pot from bottom
O/OR/R: TURN R 37 CCW → 2nd pot from bottom

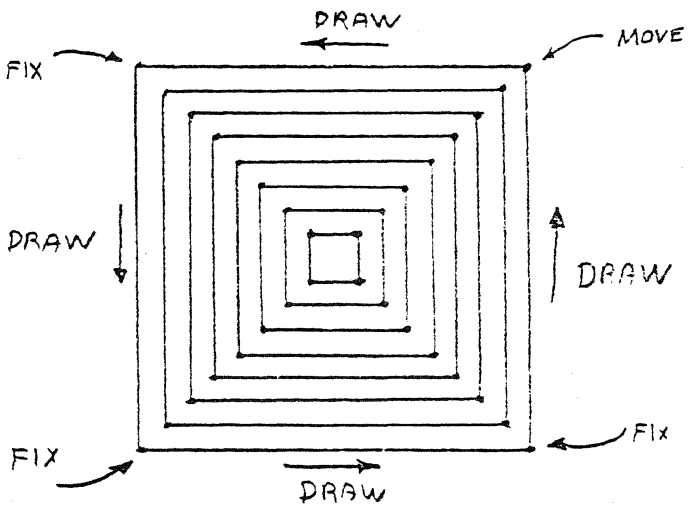
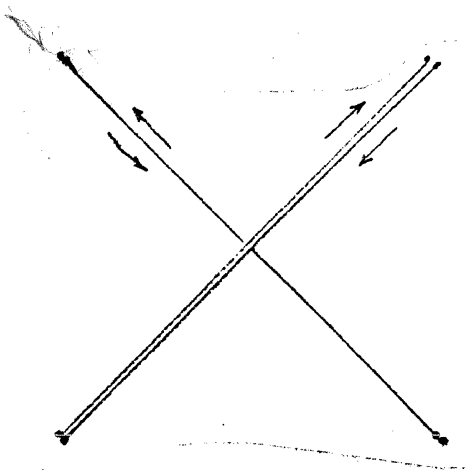


FIG. 8.8

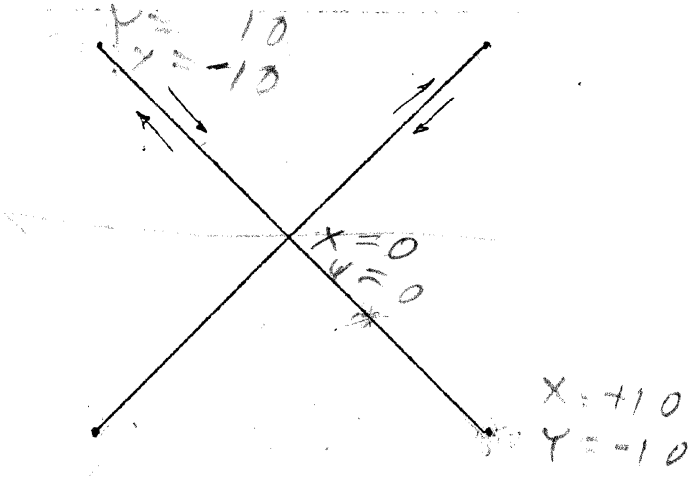
ADJUSTMENTS
CORRECT

Line Extension - off
End Points - illuminated
Display Contrast - off



POOR
Lines not overlayed

FIG. 8.9
DISPLAY PHASE ADJUSTMENT



GOOD
Line split minimumized

Adjust X Compensation capacitors C4 and C18 to obtain straight verticle lines

FIG. 8.10
X EXTEND INTEGRATOR
and
X INTEGRATOR
switching noise compensation

Adjust Y Compensation capacitors C5 and C36 to obtain straight horizontal lines

FIG. 8.11
Y EXTEND INTEGRATOR
and
Y INTEGRATOR
switching noise compensation

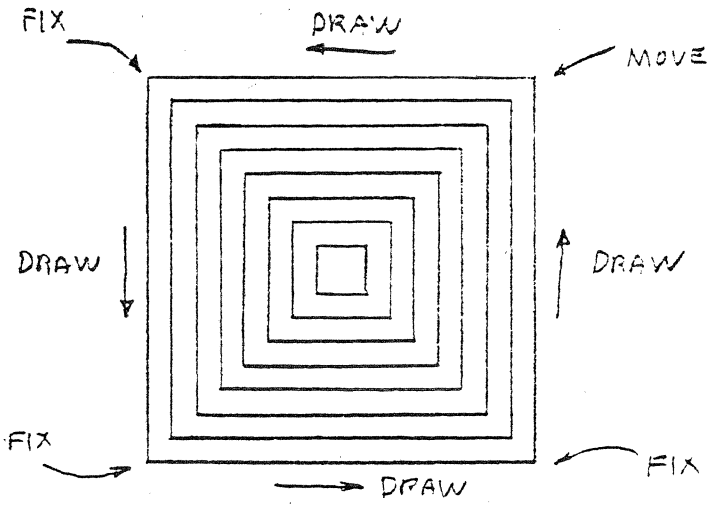
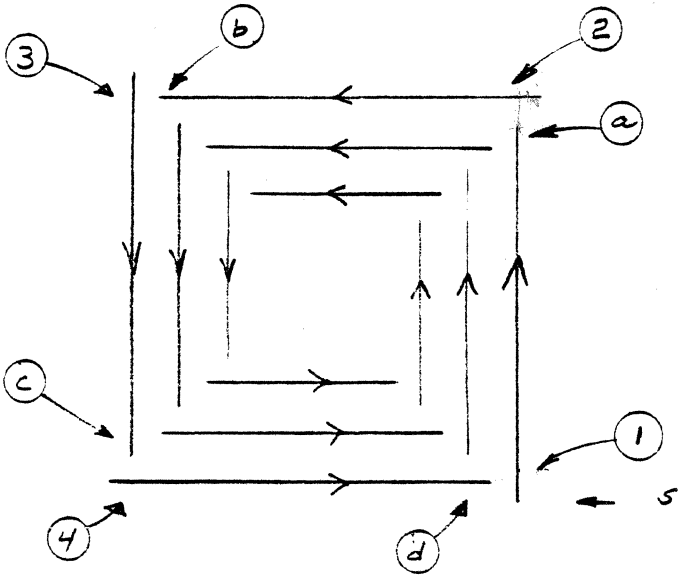


FIG. 8.12
CORRECTLY ADJUSTED
CONCENTRIC SQUARES
line extension - on
end points - blanked
- correct
delay
display contrast - off

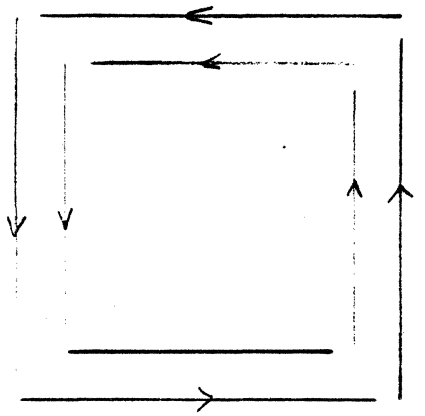
SCOPE DRIVER
END POINT MATCH

FIG. 8.13
Unblank Pulse Delay



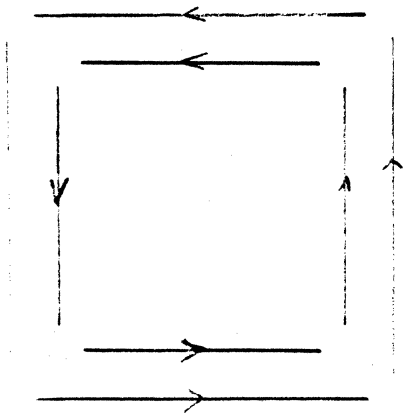
(a)
START END
NOT ENOUGH DELAY

← starts too soon



(b)
START END
CORRECT DELAY

← starts at correct time



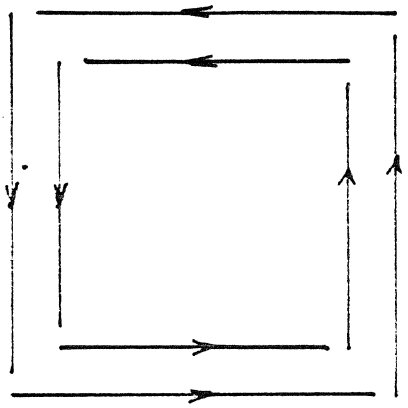
(c)
START END
TOO MUCH DELAY

← starts too late

SCOPE DRIVER
END POINT MATCH

FIG. 8.14

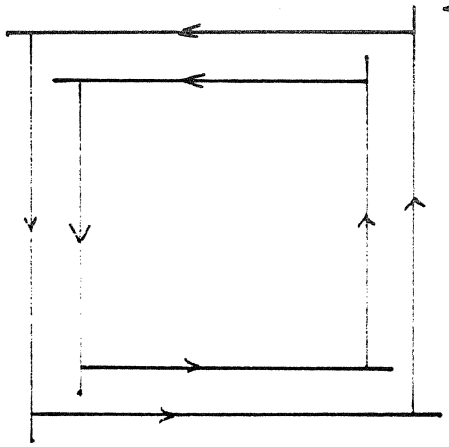
Unblank Pulse Width



← "Finnishes" too soon

(a)

FINNISH END
PULSE TOO NARROW



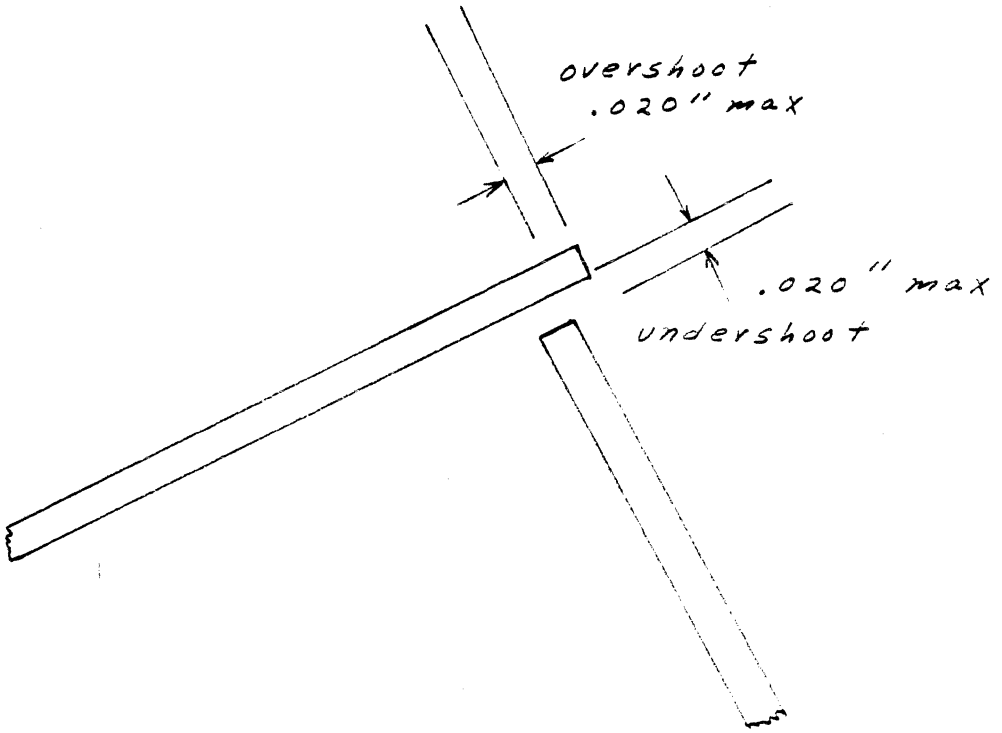
← "Finnishes" too late

(b)

FINNISH END
PULSE TOO WIDE

FIG. 8.15

END POINT
MATCH
DEFINITION



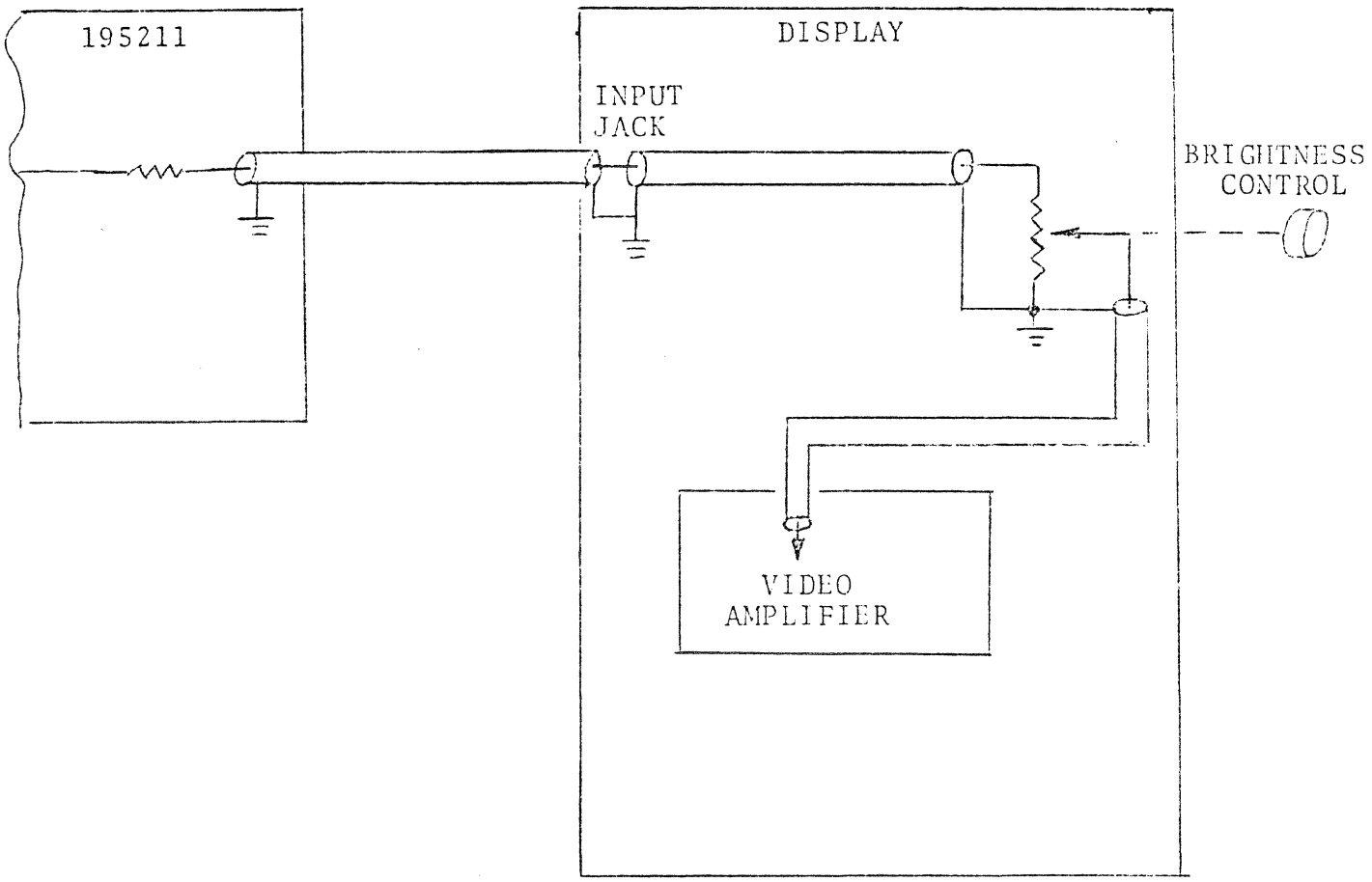


FIGURE 8.16
BRIGHTNESS CONTROL WIRING

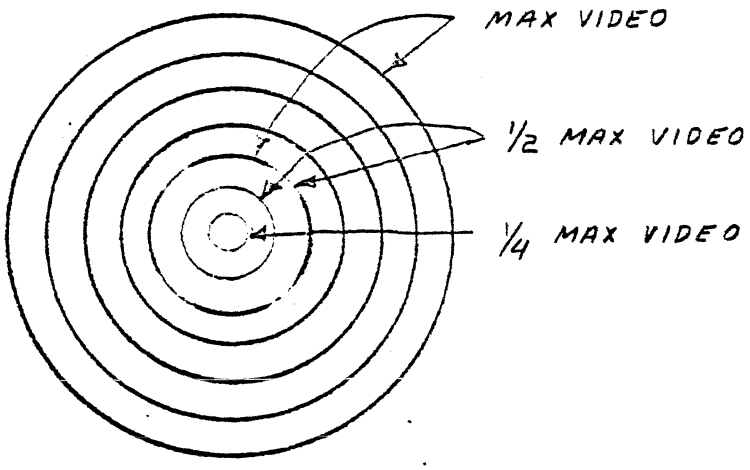


FIG. 8.17

CONCENTRIC CIRCLES

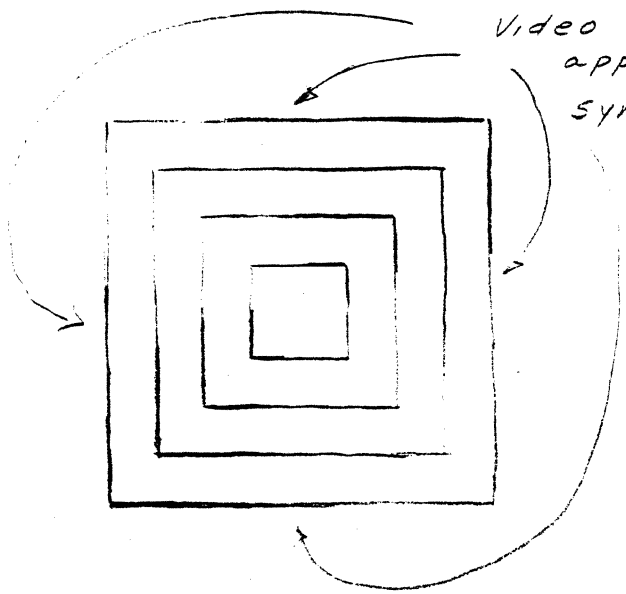
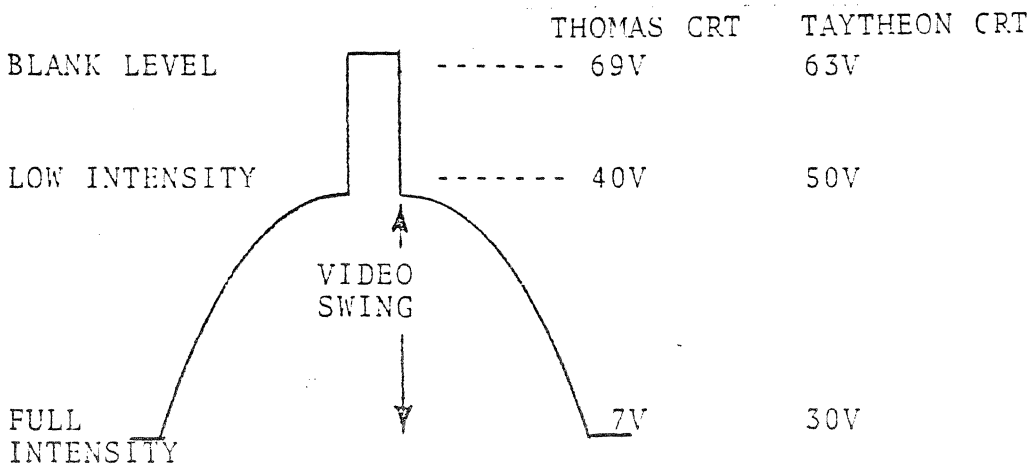


FIG. 8.18

CORRECT Z (INTENSITY) DELAY ADJUST



(CONCENTRIC BOXES PATTERN)

FIGURE 8.19

CRT CATHODE VOLTAGE WAVEFORM

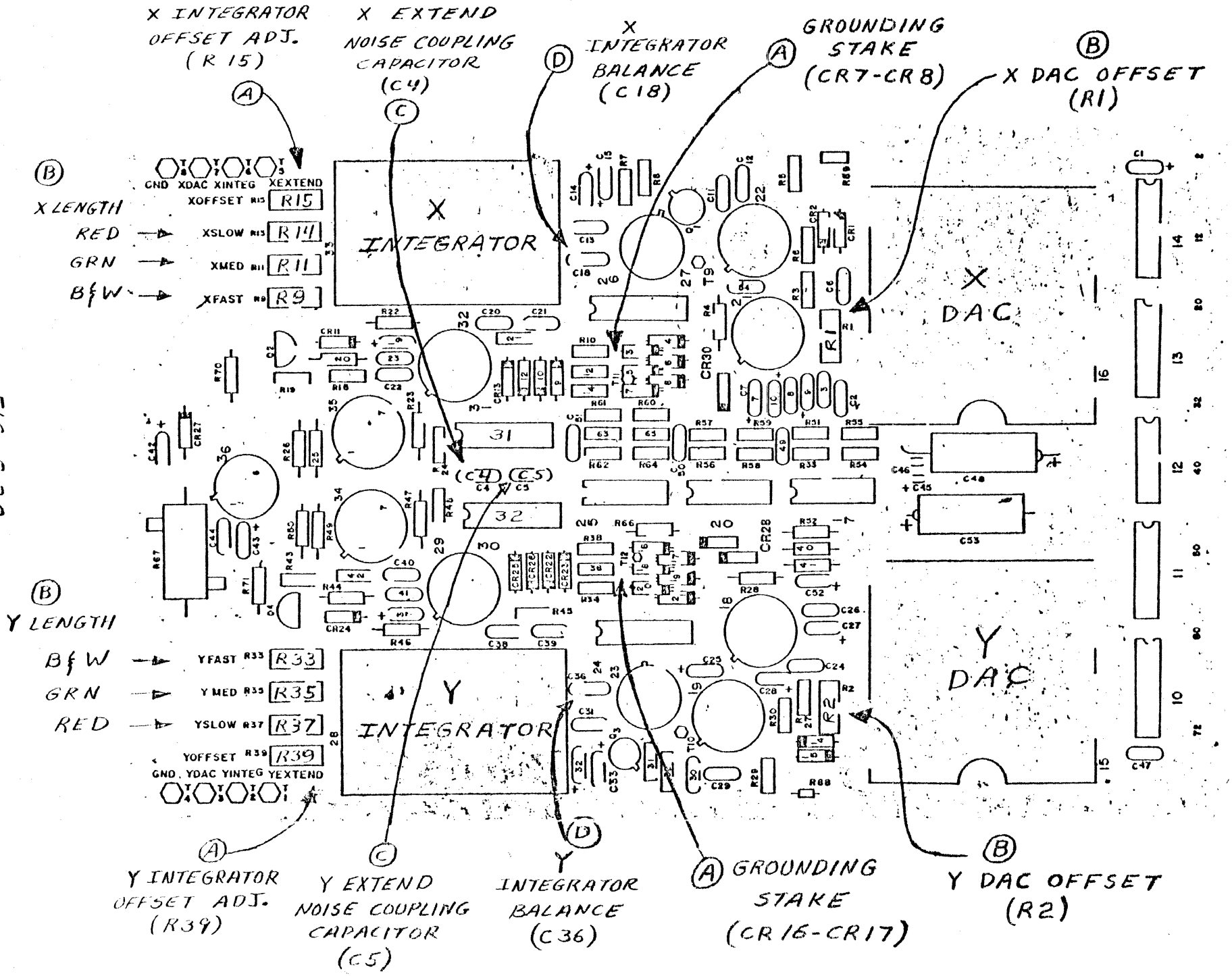


FIG. 5.20 195212 X-Y ANALOG COMPONENTS

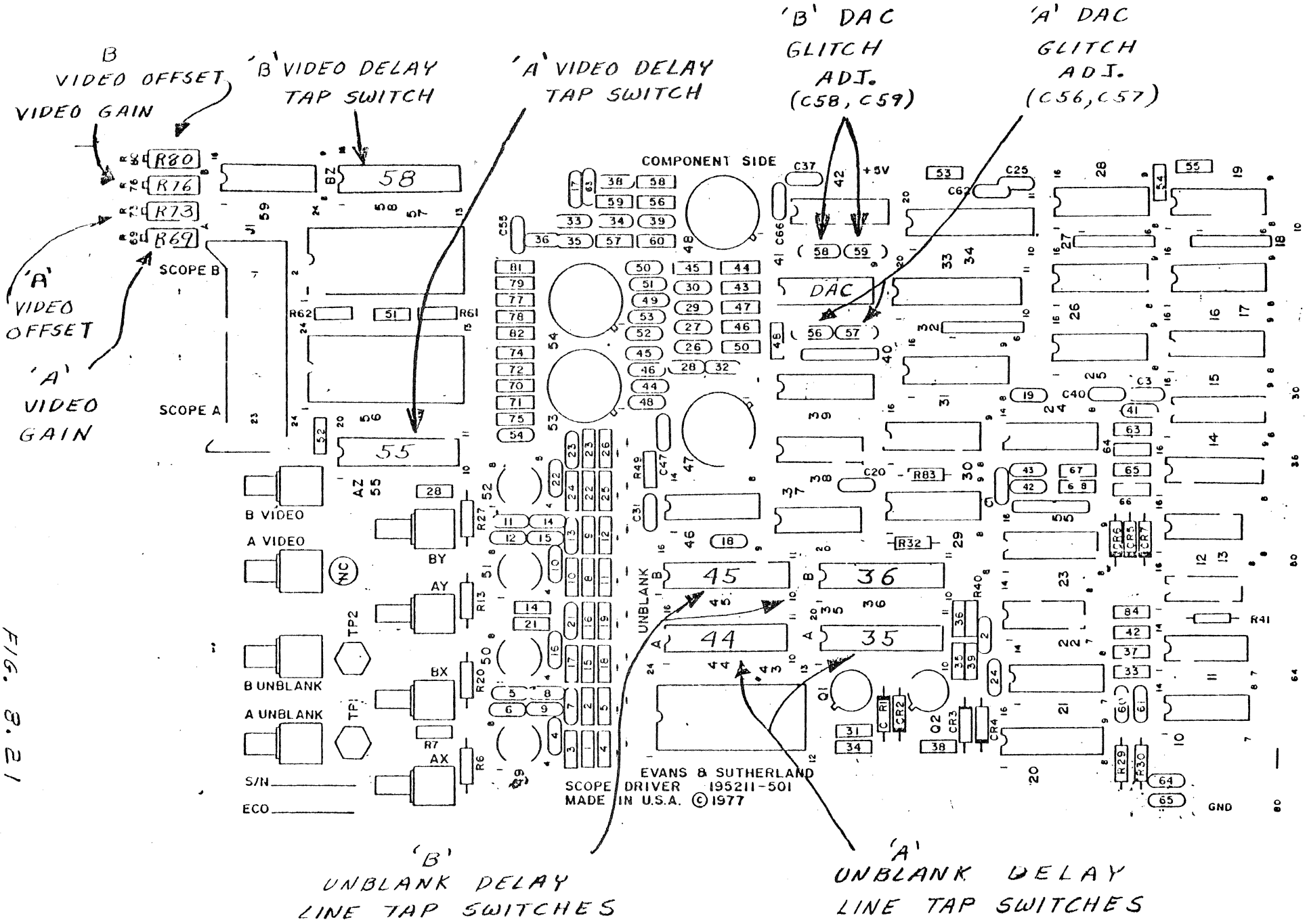


FIG. 8.21 195211 SCOPE DRIVER COMPONENTS